

国际电信联盟

ITU-T

国际电信联盟
电信标准化部门

X.1206

(04/2008)

X系列：数据网、开放系统通信和安全性
电信安全

**关于安全信息自动通知和更新发送的
厂商中立框架**

ITU-T X.1206建议书

ITU-T X 系列建议书
数据网、开放系统通信和安全性

公用数据网	
业务和设施	X.1-X.19
接口	X.20-X.49
传输、信令和交换	X.50-X.89
网络概貌	X.90-X.149
维护	X.150-X.179
管理安排	X.180-X.199
开放系统互连	
模型和记法	X.200-X.209
服务限定	X.210-X.219
连接式协议规范	X.220-X.229
无连接式协议规范	X.230-X.239
PICS书写形式	X.240-X.259
协议标识	X.260-X.269
安全协议	X.270-X.279
层管理对象	X.280-X.289
一致性测试	X.290-X.299
网间互通	
概述	X.300-X.349
卫星数据传输系统	X.350-X.369
以IP为基础的网络	X.370-X.379
报文处理系统	X.400-X.499
号码簿	X.500-X.599
OSI 组网和系统概貌	
组网	X.600-X.629
效率	X.630-X.639
业务质量	X.640-X.649
命名、寻址和登记	X.650-X.679
抽象句法记法1(ASN.1)	X.680-X.699
OSI管理	
系统管理协议子集和结构	X.700-X.709
管理通信服务和协议	X.710-X.719
管理信息的结构	X.720-X.729
管理功能	X.730-X.799
安全	X.800-X.849
OSI应用	
托付、并发和恢复	X.850-X.859
事务处理	X.860-X.879
远程操作	X.880-X.889
ASN.1的一般应用	X.890-X.899
开放分布式处理	X.900-X.999
电信安全	X.1000-

欲了解更详细信息，请查阅 ITU-T 建议书目录。

ITU-T X.1206建议书

关于安全信息自动通知和更新发送的厂商中立框架

摘要

ITU-T X.1206建议书提供了关于安全信息自动通知和更新发送的厂商中立框架。该框架的关键点是厂商中立框架。一旦注册了资产，关于漏洞信息的更新和补丁或更新即可自动提供给用户或直接提供给与资产有关的应用。

来源

ITU-T第17研究组（2005-2008年）按照世界电信标准化全会（WTSA）第1号决议规定的程序，于2008年4月18日批准了ITU-T X.1206建议书。

前言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定ITU-T各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA第1号决议规定了批准ITU-T建议书须遵循的程序。

属ITU-T研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其他一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其他机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：<http://www.itu.int/ITU-T/ipr/>。

© 国际电联 2009

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

目录

页码

1	范围	1
2	参考文献	1
3	定义	1
3.1	本建议书定义的术语	1
4	缩略语	2
5	排印惯例	2
6	安全问题介绍	2
7	当前有关漏洞信息的情况	3
8	厂商中立的概述	5
8.1	漏洞信息、更新和补丁的多个来源	5
8.2	示范应用操作	6
8.3	安全和隐私考虑	7
9	建议结构	7
9.1	消息核心层	7
9.2	消息/应用层	8
9.3	可扩容性	8
9.4	可扩展性	8
9.5	平台的独立性	9
9.6	客户端/服务器通信	9
10	框架中的组件	9
10.1	消息容器	9
10.2	版本消息	13
11	方案 (Schemas)	19
11.1	Message_Core	19
11.2	Message_Version	21
	参考文献.....	26

关于安全信息自动通知和更新发送的 厂商中立框架

1 范围

本建议书为漏洞信息的自动通知与传播的双向流动以及更新或/和补丁的发送提供了一个框架。此外，该建议书使系统管理员能够了解其职责范围内各项资产的状况。

第6和第7款从资产确认、信息传播以及系统/网络管理的角度，描述了资产维护方面出现的问题。

第8款概述的厂商中立框架中包括一个示范系统，且该框架的采用、框架运作方式和框架内和示范性交换顺序为该系统提供了支持。第8款中也描述了厂商中立的框架中应当考虑的安全问题。

第9款阐述了本建议书的功能和结构。

第10款给出了本建议书各组成部分的数据结构定义。

第11款包括第10款中定义并阐述的XML。

本建议书提供了可供任一厂商使用的框架，用于为其管辖的资产发出有关其漏洞信息的通知并接收有关其漏洞的信息，同时用于发送所需的补丁/更新，并定义了采用这一框架的各组成部分在其内部及与其它组成部分进行通信时使用的信息格式。

鉴于无需特别斟酌的情况下，许多协议便已得到支持，此建议书并未定义个部分之间进行通信时所使用的协议。

在厂商中立框架的基础上操作需要制定一些公共职能与职责，但讨论这些潜在职能的制定与实施及由此而产生的职责问题不属于本建议书的范围。

2 参考文献

没有。

3 定义

3.1 本建议书定义的术语

本建议书定义了以下术语：

3.1.1 代理：为支持某一设备上的已安装资产、服务器功能或本地服务器功能而使用本建议书的机构。

3.1.2 资产：一种设备，可根据情况分别与硬件、应用程序、操作系统或可执行代码的实例相对应。

3.1.3 客户端：需请求另一设备为其提供服务的设备。

3.1.4 设备：作为客户端、服务器或同时作为客户端和服务器、本地服务器的系统。

3.1.5 组：作为同一单元运行的一批设备。

3.1.6 本地服务器：作为补充下游客户端服务器节点的客户端。

3.1.7 信息：一种为采取具体行动提出的要求，例如，将某项资产“注册”为特定的版本和/或某些版本中的组成部分，以及“请求”获得现有或将来的更新、补丁或漏洞信息等一般性行动。扩展本建议书中功能的信息可能在本建议书以外定义。

3.1.8 信息数据：为支持某特定信息而提供的信息。在众多的可能性中，本建议书所定义的具体情况包括，定义数据的信息、与特定版本相关的漏洞信息，以及特定版本的更新与补丁。

3.1.9 信息集合：全球唯一标识符、信息及相关信息数据定义的组合与关联，均在XML方案（schema）中定义，其中XML方案源自本文定义并扩展的Message_Core。

3.1.10 补丁：为特定产品和安全漏洞而广泛发布的修正软件。一种仅对发生变化的部分进行替换而非替换整个文件的方法。

3.1.11 服务器：用于处理其它设备提出的业务请求的设备。

3.1.12 漏洞：可使计算机或计算机网络受到威胁的所有漏洞、管理程序或操作或物理缺点。

4 缩略语

本建议书使用下述缩略语和首字母缩写：

API	应用程序接口
GUID	全球唯一标识符
HTTP	超文本传输协议
ISIRT	信息安全事件响应团队
ISP	互联网业务提供商
OS	操作系统
POAS	平台/操作系统/应用/业务
URI	统一资源标识符

5 排印惯例

无。

6 安全问题介绍

随着越来越多的人开始在家庭和工作场所使用计算机，而接受过任何形式的正式计算机操作培训的用户却越来越少（更不用说与安全有关的问题），很快就出现了这样一种情况，即人们不仅几乎无法保证安全而且那些负责维护系统级安全的工作人员，在遭到破坏或发生灾难之前，越来越难以知晓其所负责并为之提供服务的系统所处的状况，而知道这一情况时却为时已晚。

造成这一问题的主要原因是由于众多计算机的维护和更新状态不同。在安全方面，系统管理已越来越是一种灾难管理和恢复程序，而不是一种预防性程序。

尽管一系列应用甚至是操作系统（OS）都拥有自身的更新机制，但它们存在着一些共性问题。其中一个问题就是首先必需启动这些更新机制，其次甚至还要在假设用户已启用通知功能的基础上，在通知用户可进行某项更新后才能进行更新。

对此，可能出现的最坏情况是这些问题使得系统管理员被完全蒙蔽，以至于不按照其职责在各个计算机上安装自我监控系统，这便造成他们对其所负责的网络和系统的总体安全性一无所知。

另一项考虑是，尽管将软件更新到最新状态至关重要，但情况通常是软件更新本身并不能解决问题，而是需要改善使用过程中的习惯做法，因此无需进行更新仅需使用最终用户收到的信息即可。但是，尽管各类应用和操作系统已有更新的机制，但就如何将持续安全使用方面的最佳做法通知用户，尚无统一的方法。

提供更新所用的方法亦十分重要。当前各类更新机制中所有更新均使用专用信道，即每次更新会话使用一条信道。但是，当为再分配中心（如互联网业务提供商（ISP）或企业网）提供了更新和其它重要信息，且在此后在网络内部以可靠且安全的方式分发这些数据包，则可有效地将分发工作所需的带宽减半或像超文本传输协议（HTTP）代理服务中的情况那样，将带宽大幅削减。

另一种在大多数情况下均得不到妥善处理的情况是，用户自己发现了一个与某给定资产的使用或操作相关的问题，却不知道应向谁咨询或向谁报告。即使已为用户和系统管理员或其他支持人员联络提供了一些方法，但使用此类系统仍然很像玩“20个问题”游戏，用户和支持人员不得不反复询问相关信息，而其回答中通常会产生更多的问题。

人们很难确切的了解大量资产的组成情况，甚至对经验丰富的专业支持人员亦是如此。由于软件使用模块化结构且各类软件实际包括来自不同厂商的多种模块（均使用其自身的版本体系），即使有关某特定产品和子模块的更新或相关信息已提供给用户，然而让用户了解更新的内容及其适用对象可能并非易事，并通常会导致信息不灵通的用户忽视其系统的安全问题。

这一情况最终将导致用户变得闭目塞听，且负责整个系统安全的人员亦被置之事外。

7 当前有关漏洞信息的情况

目前，许多厂商和大量的安全组织，例如信息安全事件响应小组（ISIRT），都在发布漏洞信息，以使用户能够认识到安全问题，并在必要时为用户提供更新和补丁。但是，最终用户通常既不使用这些信息、更新或补丁，甚至也不了解提供的这些内容是否对其适用。

造成这种情况的原因很多，但首先让我们应了解最终用户因何难以使用为其提供的信息。

最终用户一般使用资产的版本信息，来确定这些资产中是否包括会受新发现的漏洞影响的系统、软件或组件。但是，使用版本信息来检测受影响的系统、软件或组件存在困难。

- 版本标识模棱两可

各个厂家使用不同的版本标识规则。这种差异可能会造成最终用户对版本信息的解读有悖于厂家的初衷。尽管厂商使用特定的版本系统，但用户可能无法得知某更新、补丁或信息是否对其适用。

例如，假设漏洞信息包括的这一系统存在3个版本，“v1”、“v1.1”和“v1.2”，且漏洞信息如下所述：

“受影响的系统

* 受影响系统的XXXX v1版本”

可能有下列两种不同的含义：

- 1) v1仅表示v1.0版本。
- 2) v1指v1.x版本，即包括第1版的所有子版本。

将版本标识标准化是一种不切实际的方案，因此有必要阅读并理解不同厂商的版本标识规则，但是希望最终用户也能这样做很不现实。

- 某一产品的版本无法永远用作确定实际漏洞的标准。

对于一些厂商，用户通常所知道的产品版本可能无法用于检测该产品是否存在漏洞（见图1）。

大多数面向组件型的产品中存在的漏洞可能需要通过包括该产品各组件的版本来确定，而非通过产品自身的版本来确定。

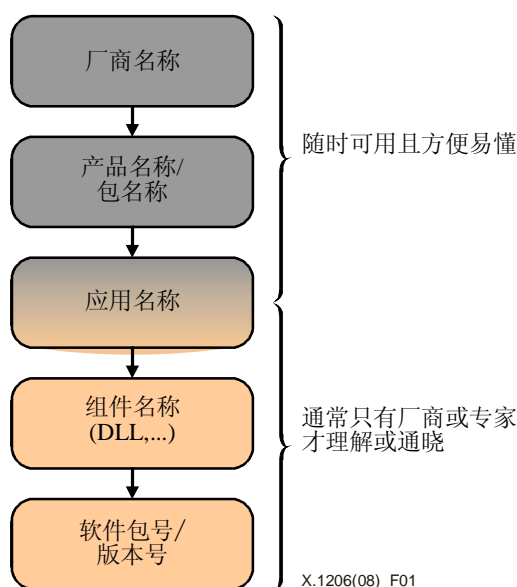


图 1 - “版本”的分层结构

- 内嵌有其它厂商产品的产品

大部分漏洞信息通常会提及某一特定产品。最终用户可能不会使用到提及的这一特定产品，但却可能使用这样一些应用，即加载或使用了存在某特定漏洞的产品业务应用。事实上，通过公开发布的应用程序接口（API），产品被用于其它应用的厂商，可能很少甚至不了解其产品被用在何处或被加载/使用在何种其它应用中。

- 系统管理员不了解其负责资产的状况

由于系统管理员要依靠用户来维护其自身的系统，并且如果用户本人不对各系统的状态进行验证或不使用用户提供的报告，则系统管理员无法了解其最终负责的系统的状态。尽管从上文有关版本的众多论述来看，用户报告功能通常既不准确亦不完整，但若没有这一功能或不存在需要开发和维护的特定监测应用，则负责维护公司或互联网业务提供商（ISP）网络安全的人员将无其它工具可用。

8 厂商中立的概述

本款概述了在厂商中立的框架下对漏洞、更新和补丁信息的分发。

通过采用这一框架，图2所示的漏洞信息更新和补丁分发系统可使用简单明了的签约方案来构建。各项资产、设备或本地服务器可在任何可用的服务器上注册，以接收请求的（拖送）或建议的（推送）漏洞信息、更新和/或补丁。

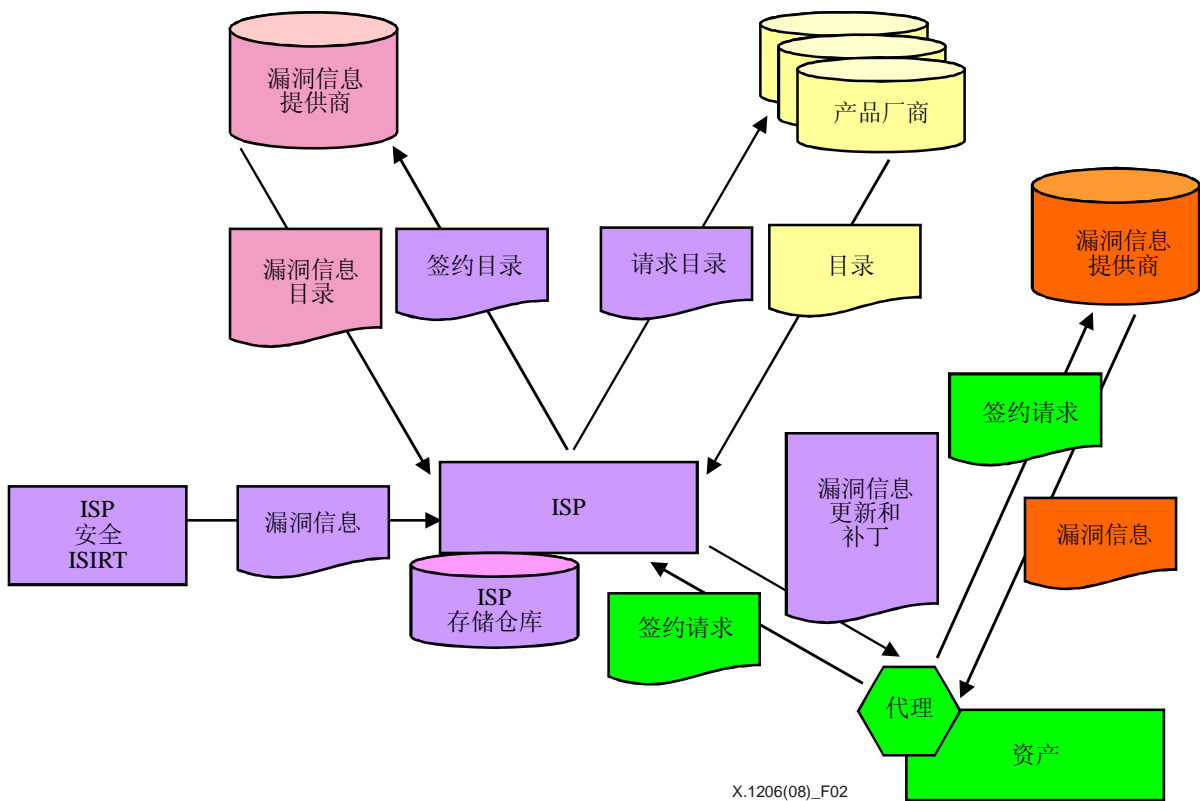


图 2 – 示范应用结构

8.1 漏洞信息、更新和补丁的多个来源

使用同一信息方案，任何实体都可以向其它实体请求和/或提供漏洞信息、更新和/或补丁。在这一具体实例中，ISP作为一个采集点成为所有用户的信息源。此外，每个单独的用户均可注册使用独立第三方的信息以获取信息更新、代码分析或者甚至是代码的更新或补丁。当然，任何希望使用某特定更新或补丁的应用都有必要根据授权接受该更新或补丁，但除使用加入签名这一得到支持的方式之外，如何实现这一操作不在本建议书的讨论范围之列。

8.2 示范应用操作

8.2.1 请求和交付信息与更新的流程

某资产（例如一个文本编辑器）已被安装且在安装的过程中通知了原已安装的代理。在资产安装过程中代理通过为其提供的信息，向用户ISP提出签约请求以获取各种漏洞信息、更新和/或补丁。

在从用户代理处收到签约信息的前后，ISP根据代理提供的有关新安装应用的信息提供商或厂商，与各类提供漏洞信息、更新和补丁的信息提供商进行联系。

此外，用户代理在这一范例中有能力分别查询其它来源的漏洞信息，并将收到的信息提供给用户察看，这一最新进程的启动可能是通过推广漏洞信息源服务的网站以及某下载链接或通过提供本建议书定义的信息（代理可据此进行操作）。

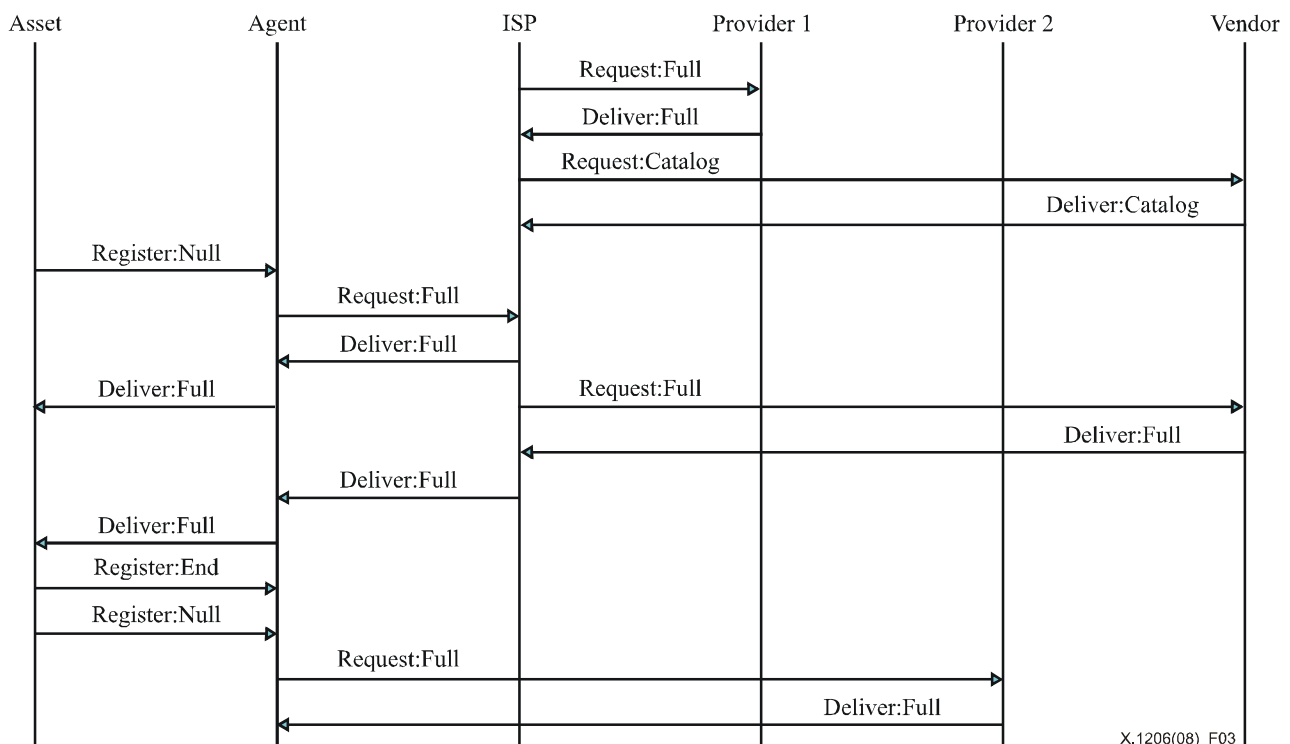


图 3 – 示范应用信息

8.2.2 用于请求和交付信息及更新的消息

ISP可以向公司网或专用网的管理员一样方便地向各漏洞信息、更新和补丁提供商提出请求。ISP可能会要求其提供所有可用的内容，以便从操作开始便作为本地存储点，或亦可请求接受有关漏洞信息、更新和补丁的可用清单（目录），并根据这一目录在今后要求提供各项要素的完整信息。尽管在此处未显示，但对完整内容请求作出的应答可以是“Major Message Not Supported”和“Minor Message Not Supported”，在这种情况下，发出请求的业务将尝试请求获得清单（目录）的版本。但是，提供商亦应将其请求政策公之于众，从而避免再进行请求协商。

对这一具体示例而言，ISP仅请求厂商提供目录并将其作为有关厂商漏洞信息、更新和/或补丁的完整交存信息这一作法，完全能够满足需求。但是，由于提出目录请求亦被看作是订购了将来所有的信息、更新和补丁，ISP可以放心的认为其代表用户向代理提交的任何请求均将获得最新版本。相反，由于漏洞信息提供商，例如1号提供商，很可能只拥有较少的在线数据，因此请求获得所有可用信息是切实可行的。

在安装资产的过程中，安装过程可能会向本地安装代理发送一条*Request:Full*（*Major:Minor*消息类型）消息，请求在所有当前和未来的漏洞信息、更新和补丁推出后，获得有关资产或其包含组件的特定版本。

此后，代理会替新安装的资产向ISP发出一条*Request:Full*消息。

鉴于本例中的ISP已在其存储单元中载有提供商1给出的漏洞信息，该ISP可以立即发出这一信息或等收到来自厂商的综合信息应答之后再行发送。

ISP在根据其存储的目录确定该厂商便是此项资产的信息源后，检查其高速缓冲存储器内是否在本地产有相关信息，并请求厂商提供可用的完整信息，然后将其传送给代理，同时可选择将这一信息暂时存储在其自身的存储单元内。

通过代理的用户接口，用户可独立的定位漏洞信息，或可使用一个用户可能已下载的、由提供商（例如提供商2）预构的请求，责成代理请求获得现已提供的目录或实际漏洞信息，或在将来可用时要求获得这些信息。

在特定的资产或其组件升级后，使用*Register:End* 取消前一版本的更新请求，并可使用*Register:null* 针对新版本提出适用信息更新的请求。同样，如果某资产被从业务中取消，亦可使用*Register:End* 取消未来的信息更新。

8.3 安全和隐私考虑

在此建议书的框架内，用户信息并未发送到厂商。但是，有必要考虑交付信息的鉴权和完整性验证，并建议用户使用下述方法：

- 通过消息中的签名确定来源。
- 代理应验证已交付消息的真实性与完整性。
- 更新应基于非个性化，以便终端用户系统的具体信息不可传送到网络中。
- 下载或在终端设备上的安装需取得终端用户同意。

9 建议结构

9.1 消息核心层

消息核心在符合标准的服务器和满足POAS要求的客户端应用之间，提供独立于POAS（平台/操作系统/应用/业务）的通信。消息核心层将如何执行某特定操作这一复杂过程提取出来，并将其放置具体的、符合消息规定的实施中，这样管理员便可将全部精力集中在执行何种操作之上。

9.2 消息/应用层

消息/应用层包括两种主要元素。第一个元素便是通常在资产或设备操作过程中使用的独立于POAS的程序或功能，例如请求漏洞信息、更新或补丁。这些操作处于更高层面，且无论采用何种系统实施方式，均适用于各类系统和功能。例如，所有系统均需要定期更新或打补丁，并要求用户能够在相关漏洞信息可用时意识到这一点。

问题是几乎每种系统在实施层面执行大同小异的高层功能时，都会使用完全不同的方法。在该建议书的构架中，主接口的创建便是在这一层面，即独立于POAS的要求与低层实施之间发生冲突的层面。

总体而言，标识符会被指配给无论系统类型如何均需执行的高层操作/程序。此后，不同类型的系统使用这些标识符在彼此之间传送指令。收到某一指令后，接收系统简单地将此指令与和特定系统处理程序相关的数据一同传送出去，从而对特定的指令作出处理。

在许多方面，这一方式与在多种不同平台上处理并提供压缩视频流或阅读此类文件相似。各类依赖于平台的视频播放器或文本文件阅读器负责按照所涉标准处理接收到的文件。

9.3 可扩容性

使用本建议书实施的系统的可扩容性，得到了拓扑服务器/客户端标识结构的支持，如下所述，该结构支持某服务器的客户端作为另一些客户端的服务器。实际上，任何服务器和客户端的拓扑分组均受到支持，这一点可用树形图中的节点加以描述。此外，根据隶属于特定客户端或组的消息捆绑传递的能力，允许任何服务器在给定客户端上的任一层面，都能够与该客户端进行通信和互动。

- 为客户端和服务器被指配了128比特全球唯一识别符（GUID）并以此作为识别的依据
- 可为客户端预指配一个GUID或在给定服务器上注册时重新指配，或同时采用这两种方式
- 根据拓扑政策，可为某客户端指配/使用不同的GUID，以便与不同的服务器互动
- 如果某客户端自身是其它客户端的服务器（本地服务器），则本地服务器可能会“隐藏”其上级服务器的真实GUID，或者可能出现完全相反的情况
- 客户端可能向一个以上的服务器报告/被指配给一个以上的服务器，其中各类服务器负责提供不同范围的业务

9.4 可扩展性

仅通过在本建议书的基础上创建的方案并扩展本建议书中的消息核心方案，便可加入新的平台、应用、功能和业务。

通过定义的消息方案，便可为希望确保全面互操作性的任意平台创建模块。

基于新消息方案的消息和消息数据结构，用户接口可以方便地按需创建。

根据需要，通过导入现有消息定义、消息数据结构或者这两者，新消息集可以“重新利用”公共消息。

9.5 平台的独立性

- 本建议书是各参与实体间的“标准化”实体。
- 使用本建议书的模块，其创建是基于根据需求，依靠POAS模块使用各类消息、功能和业务。
- 无论平台使用何种设备，任何设备均可向其它相关设备发出特定的消息或请求。在某些具体情况下，包含的消息数据可能包括针对具体平台的内容，但在任何情况下，本建议书都希望免除对跨平台知识的需求。本建议书仅支持一种设备命令另一设备应做何操作及如何操作。

9.6 客户端/服务器通信

9.6.1 公共协议

- 基于XML

9.6.2 服务器 – 客户端通信模式

- 服务器推送
 - 服务器在通知和更新可用后向已注册的客户端发送。
- 服务器拖送
 - 服务器向客户端发出操作、通知和更新请求。
- 客户端推送
 - 客户端向服务器发送通知和更新。
- 客户端拖送
 - 客户端向服务器发出操作、通知和更新请求。

10 框架中的组件

本建议书使用XML进行定义且书中包括的消息结构支持消息源和目的地的处理，以及消息及相关消息数据的承载。

本款给出了主要组件数据格式的定义：消息容器、注册消息和版本消息。

10.1 消息容器

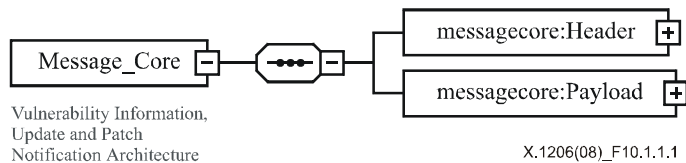
消息容器既是承载、路由和交付消息的容器，同时还作为应答和抽象的平台，为通过扩展和导出创建其它消息和应答提供场所。

10.1.1 Message_Core

本建议书结构中使用的所有消息均是通过与*Message_Core* 相关的导入、扩展和导出定义。

该消息结构的根元素为*Message_Core*。

10.1.1.1 句法



10.1.1.2 语义

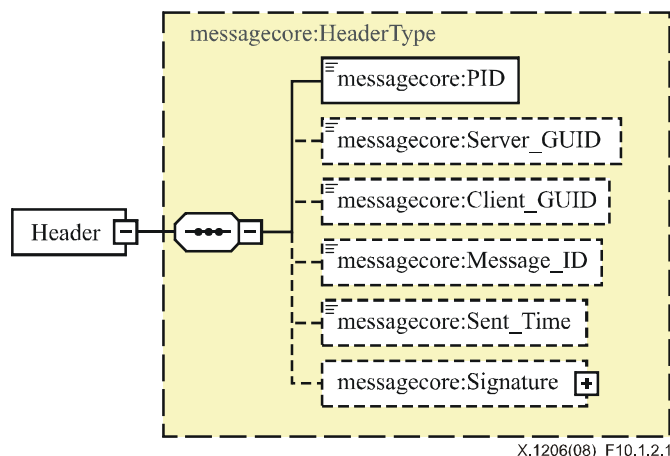
Header – 包含一般性路由和消息标识信息。[必备]

Payload – 包括XML选择的一个或多个绑定*Message_Core*消息，或单一的消息。在消息可能被缓存于特定节点（本地服务器），或（某控制/服务于本地服务器的）服务器希望收到来自本地服务器的控制/提供服务客户端的消息和消息数据时使用。[必备]

10.1.2 标题

消息的标题中包含用于扩展和互操作目的协议ID，处理某特定消息源和目的地、承载消息ID的字段、承载消息始发时间信息的字段等信息。

10.1.2.1 句法



10.1.2.2 语义

PID – 协议ID。一种128比特的值，用于确定特定的版本和/或消息协议的扩展，或进行全面的替代，其中至少要导入*Message_Core*、*Header* 和*PID*。为表明遵守本建议书，应用可使用#h00000001的值。[必备]

Server_GUID – 一种128比特的值，客户端将其用于一对一的标识客户端向其直接报告的服务器或本地服务器。[可选]

Client_GUID – 一种128比特的值，服务器和本地服务器将其用于一对一的确定向这些服务器报告的客户端。[可选]

Message_ID – 一种128比特的值，用于一对一的确定某特定消息。尽管使用该方法选择适当的*Message_ID*是一种非标准做法，但预计某特定系统的对等部分将在后续应答中使用与*Message_ID*相同的值。[可选]

Sent_Time – 包括某特定消息被发送的时间。[可选]

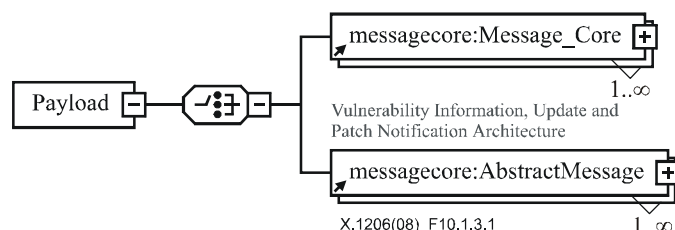
Signature – 封装的XML签名。[可选]

10.1.3 有效载荷

定义与给定*Payload*内容相关的功能，是通过定义和建立*UUID*和/或*CID* 及*Message*和与*Message*相关的*Message_Data*。*UUID/CID*、*Message*和 *Message_Data* 元素是本建议书结构中主要的可扩展点，亦是本建议书结构具有跨平台独立性的原因。已定义的*Message*及其相关*Message_Data*被称为消息集合。

通过定义以及*UUID* 和/或*CID*与消息集XML方案（其中导入并扩展了*Message_Core*方案）定义的结合，并通过针对特定消息集使用模块（其中包括*Message*和*Message_Data*方案），这些模块可以被加载至任何符合本建议书规定的的应用，同时可以确保全面的互通性。

10.1.3.1 句法



10.1.3.2 语义

Message_Core – 提供从多项资源打包并交付多条消息的能力。如果已选[必备]

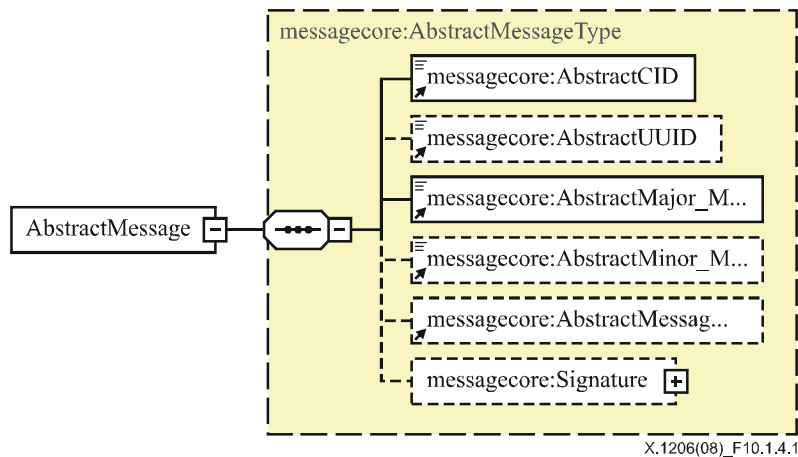
AbstractMessage – 相关的消息和模块标识符，以及特定的消息类型指示和特定的消息数据。
[必备]

10.1.4 AbstractMessage

*AbstractMessage*数据结构除用于承载确定目前究竟提出了那些消息和消息模式请求外，还承载了执行被请求消息所需的所有必要数据。

此数据结构计划通过由标准方案定义的任何特定消息扩展，且由于在核心方案中这一结构被作为抽象类别使用，因此无法在没有扩展定义的情况下加以使用。

10.1.4.1 句法



10.1.4.2 语义

AbstractCID – 一种128比特的值，用于一对一的确定某特定功能。通过给定*UUID*确定的某特定实施模块，可支持通过*CID*确定的多种功能。相反，多个模块可使用相同的功能，但用于不同的应用，例如文字处理程序版本模块和视频编辑版本模块。[必备]

AbstractUUID – 一种128比特的值，用于一对一的确定已实施的、可处理与其ID相关*Messages*模块。[可选]

AbstractMajor_Message – 一种字符串，用于携带服务器发往客户端的指令或客户端发往服务器的指令。这些指令在负责导入和扩展*Message_Core*方案中的定义，且可能包括但不限于“Register”、“Request”等。[必备]

AbstractMinor_Message – 一种补充消息标识字符串，用于进一步定义所指消息的模式。此类指令在负责导入和扩展*Message_Core*的方案中定义，且这些模式可能使用，但不限于“Full”、“Catalog”等。[可选]

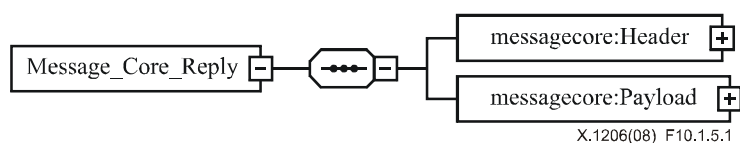
AbstractMessage_Data – 处理请求的消息所需的数据。[可选]

Signature – 封装的XML签名。[可选]

10.1.5 Message_Core_Reply

*Message_Core_Reply*数据结构用于携带状态应答以及任何其它需要的额外数据，将其送至任何发起特定通信流的实体。

10.1.5.1 句法



10.1.5.2 语义

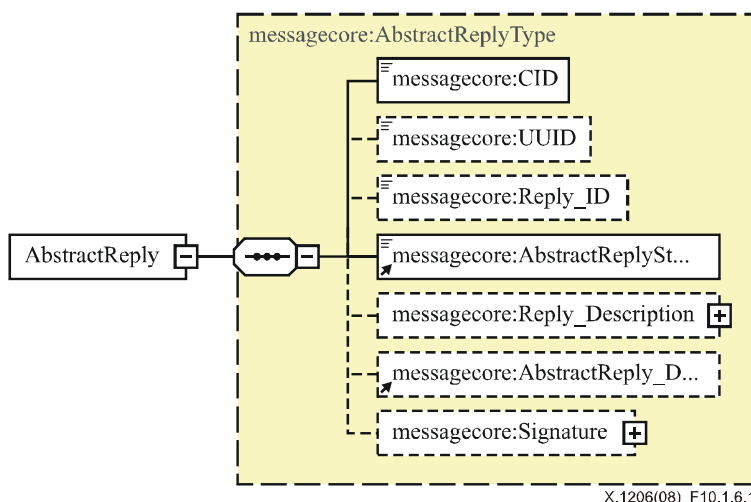
Header – 结构、数据和要求与 *Message_Core:Header* 相同。

Payload – 包括 XML 选择的一个或多个绑定 *Message_Core_Reply* 消息，或单一的 *Message_Core_Reply* 消息。在消息可能被缓存于特定节点（本地服务器），或（某控制/服务于本地服务器的）服务器希望收到来自本地服务器控制/提供服务的客户端的消息和消息数据时使用。[必备]

10.1.6 Abstract_Reply

Abstract_Reply 数据结构用于携带状态应答以及任何其它需要的额外数据，将其送至任何发起特定通信流的实体。

10.1.6.1 句法



10.1.6.2 语义

CID – 消息ID。一种128比特的值，用于确定特定的版本和/或消息协议的扩展，或进行全面的替代，其中至少要导入 *Message_Core*、*Header* 和 *PID*。为表明遵守本建议书，应用可使用 #h00000001 的值。[必备]

UUID – 一种128比特的值，用于一对一的确定一种已实的施模块，这种模块能够处理与其ID相关的应答。[可选]

Reply_ID – 一种128比特的值，用于一对一的确定某特定应答。尽管使用该方法选择适当的 *Reply_ID* 是一种非标准做法，但预计某特定系统的对等部分将在收到的请求（以此作为应答）中使用与 *Message_ID* 相同的值。[可选]

AbstractReplyString – 经引入 *Message_Core* 方案的消息扩展并包含指出前一请求（该应答所针对的请求）所处状态的字符串。此类字符串可能是，但不仅限于“Failed”、“Succeeded”、“Message not supported”等。[必备]

Reply_Description – 普通文本或HTML，进一步给出有关 *AbstractReplyString* 的信息。[可选]

AbstractReply_Data – 需要的任何与实施相关的数据。[可选]

Signature – 封装的XML签名。[可选]

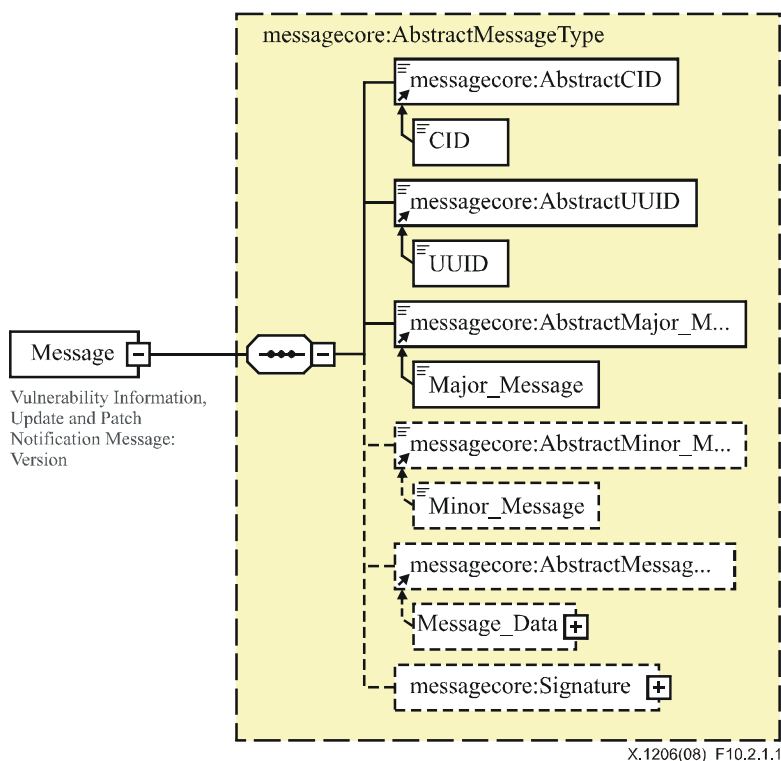
10.2 版本消息

版本消息在某系统的组成部分间使用，用于请求并提供漏洞信息以及更新和/或补丁。

10.2.1 版本：消息

*Version:Message*旨在直接替代任何存在*AbstractMessage*的位置，且其数据结构不仅受到严格限制，同时在经过扩展后成为唯一支持提供、请求和交付漏洞信息以及更新和/或补丁程序的方式。

10.2.1.1 句法



10.2.1.2 语义

CID – 用特定的指配值#h02代替并限制*Message_Core:AbstractCID*基础类别。它是唯一可确定此消息属于版本消息类别的标识。[必备]

UUID – 代替*Message_Core:AbstractUUID*消息的基本类别，并作为唯一的方式用于识别能够处理与其ID相关的*Messages*的已实施模块。[必备]

Major_Message – 使用“Register”、“Request”、“Deliver”、“Analyse”等可能值的枚举清单，替代并限制*Message_Core:AbstractMajor_Message*基本类别。[必备]

这些值的用法如下：

- “Register” – 通知接收客户端目前正在使用*Version:Message_Data*中描述的资产及其版本，并应在漏洞信息、更新和补丁可用后予以提供。
- “Request” – 为各种产品请求可用的漏洞信息、更新和补丁，最低限度的满足*Version:Message_Data*中提出的要求。
- “Deliver” – *Version:Message_Data*中包括漏洞信息、一个或多个更新或补丁。
- “Analyse” – 请求对*Version:Message_Data*中包含的信息进行分析。该请求的原因可能是演绎或在*version:Info* – *version:Description or version:Info* – *version:Container* – *version:Description* 中给出的解释，具体要根据那一方式更适合该应用而定。

Minor_Message – 使用“Full”、“Catalog”和“End”等可能值的枚举清单，替代并限制 *Message_Core:AbstractMinor_Message* 基本类别。[可选]

这些值的用法如下：

- “Full” – 请示获得所有可用和适用的更新和/或补丁。
- “Catalog” – 交付 *Version:Message_Data* 中规定产品的漏洞信息、更新和/或补丁清单。
- “End” – 根据 *Version:Message_Data* 中包含的信息结束某项签约。

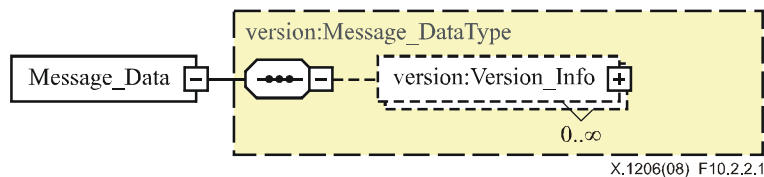
Message_Data – 通过用于携带产品版本标识以及任何适用的漏洞信息、更新和/或补丁替代并限制 *Message_Core:AbstractMessage_Data* 基本类别。[可选]

Signature – 封装的XML签名。[可选]

10.2.2 版本： Message_Data

Version:Message_Data 计划直接替代任何存在 *AbstractMessage_Data* 的位置，且其数据结构不仅受到限制而且还进行了扩展，以便为携带漏洞信息、更新和/或补丁提供独一无二的支持。

10.2.2.1 句法



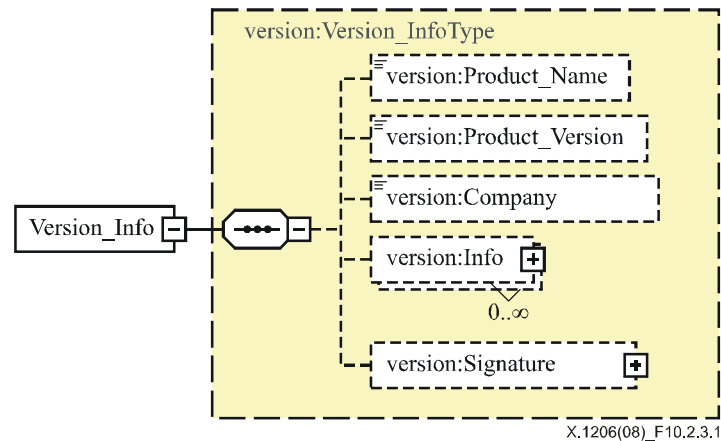
10.2.2.2 语义

Version_Info – 用于确定要求提供的具体情况等级、产品及其版本，并能够支持携带漏洞信息、更新和/或补丁的信息。

10.2.3 Version_Info

不仅用于确定产品及相应的厂商，版本和与版本相关的信息，还用于携带所需的漏洞信息、更新和补丁，以便向有需要的应用分发。

10.2.3.1 句法



10.2.3.2 语义

Product_Name – 已确定版本信息的顶级产品的名称，例如对包含有特定闪存系统版本的某特定 DVD 驱动器，*Product_Name* 将参照 DVD 驱动器的产品名。[可选]

Product_Version – 已确定版本信息的顶级产品的版本。参照前一示例，*Product_Version* 将就 DVD 驱动器的具体版本作出回答。[可选]

Company – 已确定产品或资产的厂商、提供商或作者。[可选]

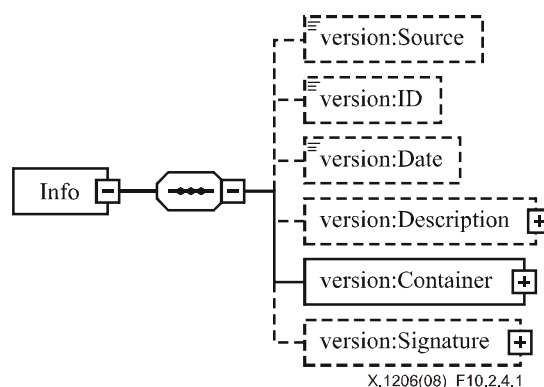
Info – 一种数据结构，如前几个字段值所示，用于携带产品或资料实际使用方面的漏洞信息。[可选]

Signature – 封装的 XML 签名。[可选]

注 – *Info* 中包含的信息总是用于确定某特定资产，其中 *Product_Name* 和 *Product_Version* 的使用仅是为了方便，但是，当 *Info* 中确定的资产为顶级产品时，建议用 *Product_Name* 和 *Product_Version* 明确指出确定的资产中不存在“parent”。

10.2.4 Info

10.2.4.1 句法



10.2.4.2 语义

Source – 一种文本类的说明，例如厂商的名称或（提供已包括的代码或漏洞信息的）漏洞信息提供商。[可选]

ID – 可能能够确定特定*Info* 数据集的ID。尽管没有确定指配的方法，但现建议指配ID在来自特定*Source*的特定资产序列内，仅需保持唯一性。[可选]

Date – 与*Info*数据相关的XML数据值。[可选]

Description – 一种文本性或HTML说明，用于描述可选*Info_Container*实例中的代码，或其中此数据结构特定实例所述特定资产版本的统一资源标识符（URI）。[可选]

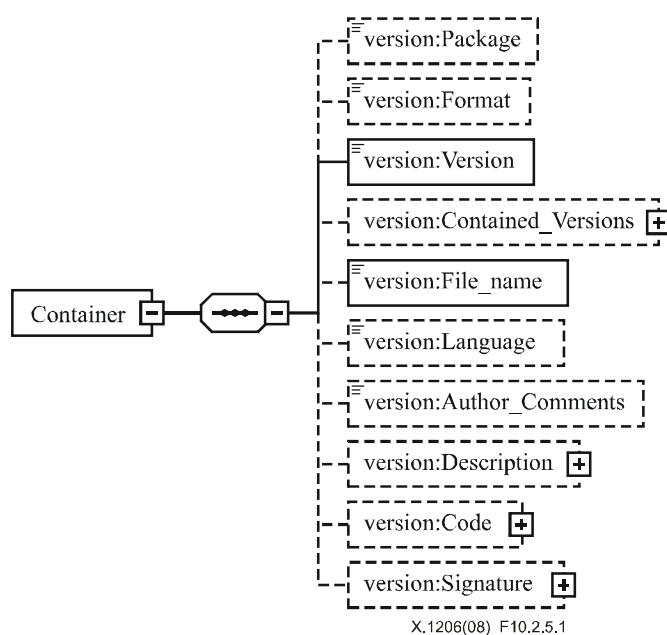
Container – 一种数据结构，用于携带更新或补丁形式的代码。[必备]

Signature – 封装的XML签名。[选可]

10.2.5 容器

承载版本信息、漏洞信息、更新或补丁的容器。

10.2.5.1 句法



10.2.5.2 语义

Package – *Code*特定版本和/或内容使用的打包方法。[可选]

Format – 特定版本的格式。[可选]

Version – 适用的版本，用于请求或确定*Version_Control* 和/或*Code*的内容。[必备]

Contained_Versions – 一类清单，通过*Version_Info*（见上文）数据结构进行定义，其结构中
包含对当前资产中所有组件或单个组件的说明。[可选]

File_Name – 文件名称。[可选]

Language – 针对不同国家存在不同版本的资产是基于语言，这一字段便是用于确定使用的语言。[可选]

Author_Comments – 所需的任何文本内容。[可选]

Description – 一种文本性或HTML说明，用于描述*Code*中包含的特定版本，或通过*Version* 和/或 *File_Name*确定的资产漏洞信息，这些信息由*Version_Info: Product_Name*和*Version_Info: Product_Version*使用或存在于*Version_Info : Product_Name*和*Version_Info: Product_Version*的内部。

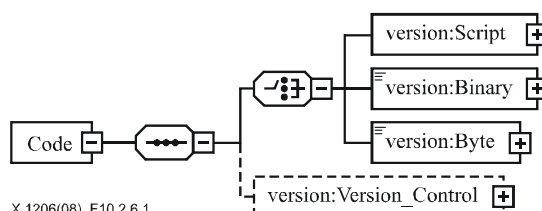
Code – 可执行代码及版本控制信息容器。

Signature – 封装的XML签名。[选可]

10.2.6 代码

实际代码或字节串容器，例如作为内存转储或相关URL参考以及可用于控制版本更新的信息。

10.2.6.1 句法



10.2.6.2 语义

Script – 脚本形态的代码。[可选择做为必备]

Binary – 汇编二进制代码。[可选择做为必备]

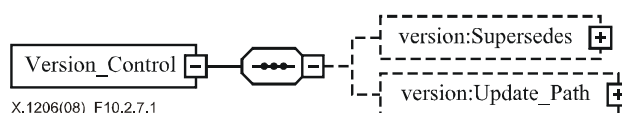
Byte – 供分析的数据。[可选择做为必备]

Version_Control – 一种数据结构，用于指出连续版本之间的关系及其可能采用的更新路径 [选可]

10.2.7 Version_Control

一种容器，包括承载用特定版本取代的、按序排列版本的清单，同时还包括指出已确定的更新或补丁适用于哪些版本的清单。

10.2.7.1 句法



10.2.7.2 语义

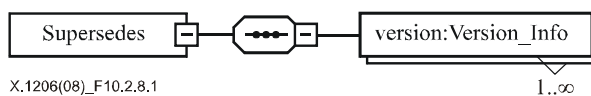
Supersedes – 由 *Version_Info* 取代的按序排列的版本清单。[选可]

Update_Path – 当前更新或补丁可能适用的版本清单。[选可]

10.2.8 取代

特定版本取代的资产清单。

10.2.8.1 句法



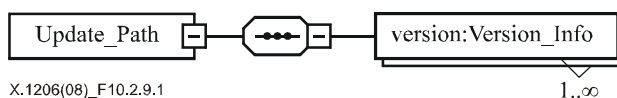
10.2.8.2 语义

Version_Info – (见 *Info*)。

10.2.9 Update_Path

包含更新或补丁可能适用的资产清单。

10.2.9.1 句法



10.2.9.2 语义

Version_Info – (见 *Info*)。

10.2.10 ReplyStatus

10.2.10.1 句法



10.2.10.2 语义

ReplyStatus – 包含下述内容之一：

“Failed” – 请求的操作无法完成。

“Succeeded” – 请求的操作获得成功。

“Major Message Not Supported” – 不支持请求的 Major Message 标识符或其某些情况下的使用。

“Minor Message Not Supported” – 不支持请求的 Minor Message 标识符或其某些情况下的使用。

“Handler not available” – 不支持确定的消息类型。

11 方案 (Schemas)

11.1 Message_Core

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:messagecore="http://www.itu.int/xml-namespace/itu-t/x.1206/CORE/"
xmlns:ns1="http://www.w3.org/2000/09/xmlsig#" targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.1206/CORE/"
elementFormDefault="qualified" attributeFormDefault="unqualified">
```

```
<xs:import namespace="http://www.w3.org/2000/09/xmlsig#" schemaLocation="xmlsig-core-schema.xsd"/>
```

```
<xs:element name="Message_Core">
```

```
<xs:annotation>
```

```
<xs:documentation>Vulnerability Information, Update and Patch Notification Architecture</xs:documentation>
```

```
</xs:annotation>
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="Header" type="messagecore:HeaderType"/>
```

```
<xs:element name="Payload">
```

```
<xs:complexType>
```

```
<xs:choice>
```

```
<xs:element ref="messagecore:Message_Core" maxOccurs="unbounded"/>
```

```
<xs:element ref="messagecore:AbstractMessage" maxOccurs="unbounded"/>
```

```
</xs:choice>
```

```

        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="HeaderType">
    <xs:sequence>
        <xs:element name="PID" type="xs:string"/>
        <xs:element name="Server_GUID" type="xs:string" minOccurs="0"/>
        <xs:element name="Client_GUID" type="xs:string" minOccurs="0"/>
        <xs:element name="Message_ID" type="xs:string" minOccurs="0"/>
        <xs:element name="Sent_Time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="Signature" type="ns1:SignatureType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="AbstractMessage" type="messagecore:AbstractMessageType" abstract="true"/>
<xs:complexType name="AbstractMessageType">
    <xs:sequence>
        <xs:element ref="messagecore:AbstractCID"/>
        <xs:element ref="messagecore:AbstractUUID" minOccurs="0"/>
        <xs:element ref="messagecore:AbstractMajor_Message"/>
        <xs:element ref="messagecore:AbstractMinor_Message" minOccurs="0"/>
        <xs:element ref="messagecore:AbstractMessage_Data" minOccurs="0"/>
        <xs:element name="Signature" type="ns1:SignatureType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="AbstractCID" type="messagecore:AbstractCIDType" abstract="true"/>
<xs:complexType name="AbstractCIDType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>
<xs:element name="AbstractUUID" type="messagecore:AbstractUUIDType" abstract="true"/>
<xs:complexType name="AbstractUUIDType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>
<xs:element name="AbstractMajor_Message" type="messagecore:AbstractMajorMessageType" abstract="true"/>
<xs:complexType name="AbstractMajorMessageType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>
<xs:element name="AbstractMinor_Message" type="messagecore:AbstractMinorMessageType" abstract="true"/>
<xs:complexType name="AbstractMinorMessageType">
    <xs:simpleContent>
        <xs:extension base="xs:string"/>
    </xs:simpleContent>
</xs:complexType>

```

```

    </xs:simpleContent>
</xs:complexType>
<xs:element name="AbstractMessage_Data" type="messagecore:AbstractMessage_DataType" abstract="true"/>
<xs:complexType name="AbstractMessage_DataType"/>
<xs:element name="Message_Core_Reply">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header" type="messagecore:HeaderType"/>
      <xs:element name="Payload">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="messagecore:Message_Core_Reply" maxOccurs="unbounded"/>
            <xs:element ref="messagecore:AbstractReply"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AbstractReply" type="messagecore:AbstractReplyType"/>
<xs:complexType name="AbstractReplyType">
  <xs:sequence>
    <xs:element name="CID" type="messagecore:AbstractCIDType"/>
    <xs:element name="UUID" type="messagecore:AbstractUUIDType" minOccurs="0"/>
    <xs:element name="Reply_ID" type="xs:string" minOccurs="0"/>
    <xs:element ref="messagecore:AbstractReplyStatus"/>
    <xs:element name="Reply_Description" type="messagecore:Description_Type" minOccurs="0"/>
    <xs:element ref="messagecore:AbstractReply_Data" minOccurs="0"/>
    <xs:element name="Signature" type="ns1:SignatureType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="AbstractReplyStatus" type="messagecore:AbstractReplyStatusType" abstract="true"/>
<xs:complexType name="AbstractReplyStatusType">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="AbstractReply_Data" type="messagecore:AbstractReply_DataType" abstract="true"/>
<xs:complexType name="AbstractReply_DataType"/>
<xs:complexType name="Description_Type">
  <xs:attribute name="ref" type="xs:anyURI" use="optional"/><![CDATA[]]></xs:complexType>
</xs:schema>

```

11.2 Message_Version

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:messagecore="http://www.itu.int/xml-namespace/itu-t/x.1206/CORE/"
  xmlns:version="http://www.itu.int/xml-namespace/itu-t/x.1206/CORE/MESSAGE/VERSION/"
  xmlns:xmldsig="http://www.w3.org/2000/09/xmldsig#" targetNamespace="http://www.itu.int/xml-namespace/itu-t/x.1206/CORE/MESSAGE/VERSION/" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.example.com/CORE" schemaLocation="Message_Core.xsd"/>

```

```

<xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-schema.xsd"/>
<xs:element name="Message" type="messagecore:AbstractMessageType" substitutionGroup="messagecore:AbstractMessage">
  <xs:annotation>
    <xs:documentation>Vulnerability Information, Update and Patch Notification Message : Version</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CID" type="messagecore:AbstractCIDType" substitutionGroup="messagecore:AbstractCID"/>
<xs:element name="UUID" type="messagecore:AbstractUUIDType" substitutionGroup="messagecore:AbstractUUID"/>
<xs:element name="Major_Message" type="version:Major_MessageType" substitutionGroup="messagecore:AbstractMajor_Message"/>
<xs:complexType name="Major_MessageType">
  <xs:simpleContent>
    <xs:restriction base="messagecore:AbstractMajorMessageType">
      <xs:enumeration value="Register"/>
      <xs:enumeration value="Request"/>
      <xs:enumeration value="Deliver"/>
      <xs:enumeration value="Analyse"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="Minor_Message" type="version:Minor_MessageType" substitutionGroup="messagecore:AbstractMinor_Message"/>
<xs:complexType name="Minor_MessageType">
  <xs:simpleContent>
    <xs:restriction base="messagecore:AbstractMinorMessageType">
      <xs:enumeration value="Full"/>
      <xs:enumeration value="Catalog"/>
      <xs:enumeration value="End"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="Message_Data" type="version:Message_DataType" substitutionGroup="messagecore:AbstractMessage_Data"/>
<xs:complexType name="Message_DataType">
  <xs:complexContent>
    <xs:extension base="messagecore:AbstractMessage_DataType">
      <xs:sequence>
        <xs:element name="Version_Info" type="version:Version_InfoType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="Reply" type="messagecore:AbstractReplyType" substitutionGroup="messagecore:AbstractReply"/>
<xs:element name="ReplyStatus" substitutionGroup="messagecore:AbstractReplyStatus">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="messagecore:AbstractReplyStatusType">
        <xs:enumeration value="Failed"/>
        <xs:enumeration value="Succeeded"/>
        <xs:enumeration value="Major Message Not Supported"/>
        <xs:enumeration value="Minor Message Not Supported"/>
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>

```

```

        <xs:enumeration value="Handler not available"/>
    </xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="Reply_Data" type="messagecore:AbstractReply_DataType" substitutionGroup="messagecore:AbstractReply_Data"/>
<xs:complexType name="Version_InfoType">
    <xs:complexContent>
        <xs:extension base="messagecore:AbstractReply_DataType">
            <xs:sequence>
                <xs:element name="Product_Name" type="xs:string" minOccurs="0"/>
                <xs:element name="Product_Version" type="xs:string" minOccurs="0"/>
                <xs:element name="Company" type="xs:string" minOccurs="0"/>
                <xs:element name="Info" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Source" type="xs:string" minOccurs="0"/>
                            <xs:element name="ID" type="xs:string" minOccurs="0"/>
                            <xs:element name="Date" type="xs:string" minOccurs="0"/>
                            <xs:element name="Description" minOccurs="0">
                                <xs:complexType>
                                    <xs:attribute name="ref" type="xs:anyURI" use="optional"/><![CDATA[]]></xs:complexType>
                                </xs:element>
                                <xs:element name="Container">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="Package" type="xs:string" minOccurs="0"/>
                                            <xs:element name="Format" type="xs:string" minOccurs="0"/>
                                            <xs:element name="Version" type="xs:string"/>
                                            <xs:element name="Contained_Versions" type="version:Version_InfoType" minOccurs="0"/>
                                            <xs:element name="File_name" type="xs:string" minOccurs="0"/>
                                            <xs:element name="Language" type="xs:string" minOccurs="0"/>
                                            <xs:element name="Author_Comments" type="xs:string" minOccurs="0"/>
                                            <xs:element name="Description" type="messagecore:Description_Type" minOccurs="0"/>
                                            <xs:element name="Code" minOccurs="0">
                                                <xs:complexType>
                                                    <xs:sequence>
                                                        <xs:choice>
                                                            <xs:element name="Script">
                                                                <xs:complexType>
                                                                    <xs:attribute name="ref" type="xs:anyURI" use="optional"/>
                                                                </xs:complexType>
                                                            </xs:element>
                                                            <xs:element name="Binary">
                                                                <xs:complexType>
                                                                    <xs:simpleContent>
                                                                        <xs:extension base="xs:base64Binary">
                                                                            <xs:attribute name="ref" type="xs:anyURI" use="optional"/>
                                                                        </xs:extension>
                                                                    </xs:simpleContent>
                                                                </xs:complexType>
                                                            </xs:choice>
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```



```
</xs:extension>  
</xs:complexContent>  
</xs:complexType>  
</xs:schema>
```

参考文献

- [b-ITU-T X.667] Recommendation ITU-T X.667 (2004) | ISO/IEC 9834-8:2005, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 object identifier components.*
- [b-IETF RFC 3023] IETF RFC 3023 (2001), *XML Media Types.*
<<http://www.ietf.org/rfc/rfc3023.txt?number=3023>>
- [b-IETF RFC 3075] IETF RFC 3075 (2001), *XML-Signature Syntax and Processing.*
<<http://www.ietf.org/rfc/rfc3075.txt?number=3075>>
- [b-XML Datatypes] W3C Datatypes:2001, *XML Schema Part 2: Datatypes*, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/RECxmlschema-2-20010502/>.
- [b-XML Signature] W3C Signature Schema:2001, *XML Signature Schema*, W3C Recommendation, Copyright © [1 March 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/xmlsigcore/xmlsig-core-schema.xsd>.
- [b-XML Structures] W3C XML Schema Part 1:2001, *XML Schema Part 1: Structures*, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/RECxmlschema-1-20010502/>.

ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其他多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其他组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	数据网、开放系统通信和安全性
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	电信系统使用的语言和一般性软件情况