基于安全部件间互动的一种混合式解析器设计

Design a composite parser based on security components exchange

侍伟敏 杨义先 褚永刚 杨亚飞 （北京邮电大学信息安全中心 100876）

摘要：在大规模分布式入侵检测系统中，为了实现安全部件高效访问 XML 文档消息和处理数据的能力、以及提高安全部件间互动的效率，本文首先通过对三种 XML 文档解析方式分析比较之后，提出了一种解决方案，即设计了一种混合式解析器，然后对此解析器做了详细的分析和研究。

关键词：大规模分布式 IDS 安全部件 混合式 解析器

1 引言

在大规模分布式入侵检测系统中，为了实现 IDS 内部各组件和不同安全部件之间的互联互动、信息共享而采用了由 XML 定义的消息进行通信，然而 XML 文档本身只不过是一个文本文件，它并不带有任何的处理数据的能力，因此为了保证各个部件之间进行通信并能理解对方发来的信息，必需设计一个能够识别 XML 文档信息的语法解析器(parser)来解析 XML 文档并提取其中的信息。目前存在有三种访问 XML 数据的方式，分别是 DOM 方式、SAX 方式、数据绑定方式。由于不同解析程序实现的方式不同，对数据访问和处理的能力也不同。以至于采用不同的技术实现方案会适合不同的问题域。所以我们就需要设计一种适合于本系统的解析方案来实现安全部件之间交换的 XML 文档信息的解析技术。

2 XML 文档的解析技术

目前对 XML 文档的解析有三种方式，分别是 DOM 方式、SAX 方式、数据绑定方式。其中微软提供了两种进行 XML 文档解析的接口，一种是基于文档对象模型(DOM)，另外一种是 SAX(simple API for XML)。如果安装了 IE5.0 以上版本的话，计算机系统中就存在了这两种 XML 文档语法解析器。这两种解析器实际上是一个 COM 对象库，里面封装了所有进行 XML 解析需要的所有必要的对象。再一种是数据绑定方式，其实质是从 Java 应用程序内部访问 XML 数据的一种新方法。

2.1 DOM 方式

DOM (Document Object Model)是 W3C 发展的访问 XML 文档的一种标准 API。当使用 DOM 对 XML 文本文档进行操作时，它首先要解析文档，将文档中的元素、属性、注释、处理指令等都看作节点，然后在内存中以节点树的形式创建 XML 文档表示。总之，采用 DOM 对 XML 文档进行解析时，必须先在内存中生成 DOM 树，此后，应用程序就可以通过节点树访问文档的内容。

DOM 解析技术的树型结构思想与 XML 文档的结构相吻合，它把整个 XML 文档以一棵树的形式存放在内存中，通过树结构应用程序可以很容易实现随机访问。同时，这种访问方式给应用程序的开发带来了很大的灵活性，它可以任意地控制整个 XML 文档中的内容。然而，当 XML 文档比较大或者文档结构比较复杂时，对内存的需求就比较高，使得访问数据效率会很低。比如，当对树结构执行转换时，系统可能停止运转甚至彻底崩溃。

2.2 SAX 方式

SAX (Simple API For XML)是一种基于事件的 XML 文档解析标准，它是通过回调机制实现对数据的操作而不必读入整个 XML 文档到内存中。当解析 XML 文档时，SAX 模型中指定的某些事件将触发回调方法，应用程序通过实现句柄来响应不同的事件，比如文档开始事件、元素开始事件等等。

基于事件驱动 SAX 的解析技术提供的是一种对 XML 文档的顺序访问机制，对内存的要求比较低，程序执行效率高。由于基于事件驱动本身是有时序性的，对于已经分析过的部分，不能回溯重新处理。此外，与基于树结构的解析技术相比，目前支持基于事件驱动的解析器只能读 XML 文档，不能写 XML 文档，对 XML 文档的处理也缺乏一定的灵活性。

2.3 数据绑定(data binding)方式

数据绑定是 XML 文档模型的强大替代方案。尽管文档模型和数据绑定都在内存中构建文档表示，同时内部表示和标准的文本 XML 之间可以互相转换，但两者之间的不同是文档模型尽可能接近地保存 XML 结构，而数据绑定只关心应用程序使用的文档数据。因此数据绑定提供了一种简单的处理 XML 数据的方法，它主要是利用 Java 集合类处理 XML 数据。在该方法中，首先根据 XML 文档的结构构造相应的类，再构造容纳该类对象的自定义的集合类。即实现 XML 文档数据到对象实现的转换。这样做的优点在于：

(1) 将 XML 文档中存储的数据转化到用面向对象语言设计的程序中,借助面向对象程序设计语言的强大数据控制能力来对数据进行进一步的处理。

(2) 它非常便于直接访问所需的参数,而无须使用更复杂的树状结构。并且当在 XML 文档中存储配置信息时特别有用。

(3) 避免了用 DOM 解析 XML 文档时,对内存的长期占用。

(4) 避免了用 SAX 解析 XML 文档时,反复解析的过程。

然而,采用数据绑定方式对内存同样要求也很高,因此它不如 SAX 方式对 XML 文档解析的效率高。

3 基于安全部件间互动的一种混合式解析器设计

在大规模分布式入侵检测系统中,安全部件之间互动的主要目的是任何一个部件发现有入侵现象时,应及时的向其他安全部件报警并对入侵做出及时的处理。因此,这就要求安全部件在收到另一方发来的互动信息后必需以最快的速度对 XML 文档消息进行解析,获取其中的信息从而对入侵做出相应的处理。此外,就安全部件本身来说,在解析的过程中不能影响系统的性能,否则会破坏整个检测系统的检测效率。从以上两个方面来考虑,选择 SAX 的解析技术应该是一种最佳的方案。然而,SAX 对文档的处理缺乏一定的灵活性。因此,就需要设计一种适用于本系统的混合式解析器,即采用多种技术,发挥它们各自的优势,弥补其不足。

本文所设计的混合式解析器是采用 SAX 解析技术和对象层次上 XML 数据绑定机制(即 Java 集合类的绑定)相结合的方法。混合式解析器的设计模型如图 1 所示:

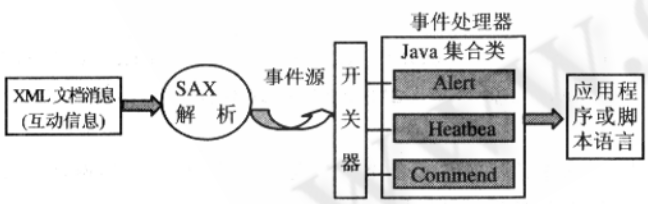

图 1　混合式解析器的设计模型

首先用 SAX 解析器对 XML 文档消息进行解析产生事件源,然后开关器再根据已经定义的规则分析事件源,不同的事件源则会驱动事件处理器相应的处理方法,于是事件就获得了处理。当然,在事件源激活事件处理器中特定方法时,会通过开关传递给事件处理器相应事件的信息,这样事件处理器才能够根据事件信息来选择相应的类函数。

3.1 SAX 解析器

SAX 解析器主要包括三个接口,分别是和 XML 文档内容相关的事件(ContentHandler)、和 DTD 相关的事件(DTDHandler)、出现错误时发生的事件(ErrorHandler),其中 ContentHandler 是实现了 SAX 应用所需要的最重要的接口,通过此接口可以报告基本解析事件及文档事件,包括文档开始事件、元素开始事件等,这将导致产生了不同的事件源。

基于 ContentHandler 接口的 SAX 解析器如图 2 所示:

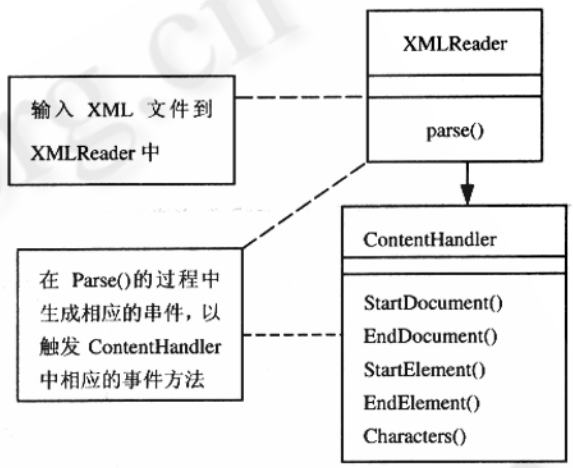

图 2　基于 ContentHandler 接口的 SAX 解析器

解析开始之前,需要向 XMLReader 注册一个 ContentHandler,也就是相当于一个事件监听器。在 ContentHandler 的接口中定义了很多方法,比如 StartDocument()、StarElement()等,因此在解析的过程中,当 XMLReader 读到合适的内容时,就会通过接口中的方法产生事件源,然后再通过开关器触发处理器中相应的响应事件。

3.2 开关器

开关器的工作是把产生的每个事件源切换到处理其事件的响应函数中,它所做的是维护一组哈希表(Hashtable)形式的规则。规则集合按照事件类型索引。当解析器报告一个事件产生时,开关器将会根据事件的类型激发事件处理器中的响应函数。

3.3 事件处理器

事件处理器是由 Java 的集合类所组成,它是遵循对象层次上 XML 数据绑定机制来实现由 XML 文档消息到 Java 集合类影射,即把 XML 源数据的描述和操作转换成面向对象类中对象属性和方法。在大规模分布式入侵检测系统中,部件间互动的消息是由 SCIMF 数据模型定义。在此模型中,根据部件间可能发生的响应类型,定义了报警 Alert、心跳 Heartbeat、命令 Command 三种类型,每种类型中又包含了它们各自的聚合类以及属性等。因此由 XML 语言所表达的 SCIMF 数据模型层次化地描述了一类 XML 文档中数

据的结构关系和类型信息。这些约束条件在 XML 数据绑定机制作用下可以转换成一个或多个类，这些类之间的关系正是约束关系的一个反映。XML 数据绑定的主要原则如下：

(1) XML 方案中特定类型定义转换为类；

(2) XML 方案中特定类型定义中的属性值及相应属性类型转换为该类型定义所转换成的类中成员变量及其类型说明；

(3) XML 方案中特定类型定义转换成的类中的成员函数为该类属性值的 set,get 操作并可再根据应用逻辑语义确定其他操作；

(4) XML 方案中某种特定类型定义中的属性值转换为相应的命令函数。比如在 Command 类型中的 name 属性，它的取值有阻断、连接、通知等命令。在转换时就需要根据属性值来定义命令函数。

(5) XML 方案中嵌套其他类型定义的特定类型定义转换为包含或引用被嵌套类型所转换的类的类。

XML 数据绑定处理遵从以上主要原则，递归实现 XML 模式到类层次的影射，进而实现 XML 文档数据到对象实现的转换。元素转变为对象，对它的每个属性进行检查，如果属性是一个特征，就为此对象创建一个简单的基本类型成员变量；若属性是元素，则创建一个新的对象，并作为一个成员变量将其添加，然后在这个新对象上又开始同样的过程，直到全部类都已创建为止。通过将 SCIMF 数据模型转换成 Java 集合类，使得事件处理器不仅能及时的处理事件源，而且还可以对它做进一步的操作。

4 结束语

本论文所设计的混合式解析器不仅解决了安全部件如何高效访问 XML 文档消息的方法，而且在很大程度上提高了对数据的处理能力，从而提高了安全部件间互动的效率。

参考文献

1 Didier Martin 等，XML 高级编程[M]，机械工业出版社 2001。

2 沈晨鸣、陈建红，基于 DOM 的 XML 数据访问技术[J]，南京工程学院学报，Vol.2，No.4，2002.11.page(s):33－35.

3 李青山、陈平，对象层次上的 XML 数据绑定模型的研究，西安电子科技大学学报(自然科学版)[J]，Vol.28，No.6，2001.12.page(s):768－771.

4 周筱媛，用 Java 集合类处理 XML 文档，西安科技学院学报[J]，Vol.22，No.3，2002.9.page(s):318－320.

5 Simeoni，F.；Lievens，D.；Conn，R.；Mangh，P.；Internet Computing，IEEE ，Volume：7 Issue：1 ，Jan.－Feb. 2003 Page(s)：19 － 27.

6 C.. Bourges－Waldegg，D.. Hild，S.G.；Generation of Java Beans to access XML data Binding，Web Information Systems Engineering，2000. Proceedings of the First International Conference on ，Volume：2 ，19－21 June 2000 Page(s)：143 －149 vol.2.

7 杨义先、孙伟、钮心忻，现代密码新理论，科学出版社，2002 年 8 月，第一版。

8 杨义先、钮心忻，网络安全理论与技术，人民邮电出版社，2003 年 10 月，第一版。