XML 在入侵检测规则中的应用研究①

Application and Study of XML in Intrusion Detection Rule

陶利民　(杭州师范学院,杭州　310036)
廖新飞　(温州职业技术学院 325035)

摘要:入侵检测系统作为一种主动安全防御系统,逐渐发展成为保障网络系统安全的重要部件。根据检测分析方法,入侵检测技术可分为异常检测方法和误用检测方法。基于规则的检测是误用检测的一种方法,而用 XML 来描述规则,可实现入侵检测系统各部件之间所交换数据形式的标准化,是 IDS 系统实现互为通用,不同入侵检测系统之间实现互操作成为可能的一种途径。

关键词:入侵检测系统　安全　规则　XML

1　入侵检测系统简介

为了保护计算机系统资源,依靠传统的被动防护是不够的。入侵检测作为新一代网络安全保障技术,提供了对内部攻击、外部攻击和误操作的实时保护,在网络系统受到危害之前拦截和响应入侵,弥补了传统安全技术的不足,逐渐发展成为保障网络系统安全的重要部件。入侵检测是通过从计算机网络或计算机系统中的关键点收集信息并对其进行分析,从中发现网络或系统中是否有违反安全策略的行为和遭到袭击的现象的一种安全技术。进行入侵检测的软件与硬件的组合便是入侵检测系统(IDS: Intrusion Detection System)。

2　两种典型的入侵检测方法

根据入侵检测分析方法所划分的异常检测方法和误用检测方法,是两种典型的入侵检测方法。误用检测就是将收集到的信息与已知的网络入侵和系统误用规则库(规则库:将已知的攻击行为进行特征提取,把这些特征用脚本语言等方法进行描述后放入规则库中)进行比较,如果其中有匹配的规则,则发生违背安全策略的入侵行为。异常检测对正常的用户行为建立统计模型,当检测对象与正常行为轮廓的偏差超过一定程度时,IDS 就会判断有入侵发生。

3　入侵检测规则

模式匹配方法是误用检测技术中的一种方法。模式匹配方法就是通过分析已知的攻击方式,找出它们的特征、产生的事件等信息,并对其进行描述,建立相应的入侵规则库,然后把收集的数据同其进行匹配。如果匹配成功,说明该数据包中含有入侵信息,计算机系统或网络受到攻击,否则属于正常行为。

模式匹配方法实现过程中针对每一种入侵行为,都提炼出它的特征值,并按照规范写成检测规则,从而形成一个规则数据库。模式匹配方法的检测过程如下:

(1) 从规则库中读取规则;

(2) 对规则进行解析,从中提取规则头和规则选项部分;

(3) IDS 从网络中抓取数据包,调用数据包解析函数,根据数据包的类型和所处的网络层次,对数据包进行协议解析,包括数据链路层、网络层和传输层;

(4) 将解析好的数据包的地址项与规则头部比较,如果相等则使用规则选项对数据包进行检查;

(5) 如果对所有规则中有一条规则检测满足,则按照规则所定义的操作方法进行报警响应;否则转到(3)继续抓取下一个数据包进行检测。

Snort 是一个功能强大、跨平台、轻量级的网络入

① 杭州师范学院科研基金项目

侵检测系统。Snort 也是一种基于规则检测的入侵检测工具。Snort 使用了一种简单但是灵活、高效的规则描述语言来对规则进行表述。以下是一个 ping 攻击扫描规则的实例：

alert icmp $ EXTERNAL_NET any - > $ HOME_NET any(msg:"ICMP webtrends scanner";content:"|00 00 00 00 45 45 45 45 45 45 45 45 45 45 45 45|";itype:8;icode:0;reference:arachnids,307;classtype:attemped - recon;sid:476;rev:1;)

含义为：ICMP 的 ping 报文中的数据含有字符串"00 00 00 00 45 45 45 45 45 45 45 45 45 45 45 45"时，向管理员发出警报。

4 XML 在检测规则中的应用

XML(eXtensible Markup Language)是一种具有数据描述功能、高度结构性及可验证性的语言。XML 是一种元标记语言，它最大的特点在于 XML 的标记和属性允许用户自行定义，并可以依照所定义的标记与属性的语法来开发应用程序。XML 文档清晰易读、易于创建。另外，还可以使用 DOM(org.w3c.dom)很方便地解析 XML 文档。Snort 规则文件使用的是一般的文本文件，对于一般 IDS 用户来说，不直观，也不易理解。而采用 XML 来描述规则，就可以很好地利用其优点，使规则直观、易理解。另外，也可以实现不同入侵检测系统之间的互操作。

4.1 规则的描述

规则采用 XML 文件来表示。一个 XML 文件就是一个规则数据库。系统的规则可以划分为两个逻辑部分：规则头(Rule Header)和规则选项(Rule Options)。规则头包含了规则动作、协议、IP 源地址和目的地址，以及源端口和目标端口值等信息。而规则选项则包含警报信息以及用于确定是否触发规则响应动作而需检查的数据包区域位置信息。

规则文件的基本结构如下：

```
<rulelist>
  <rule>
    <head attribute - list> </head>
    <option attribute - list > </option>
  </rule>
  ……
  <rule>
    <head attribute - list > </head>
    <option attribute - list > </option>
  </rule>
</rulelist>
```

标记 <rulelist>、<rule>、<head> 和 <option> 是自定义的标记。<rulelist> </rulelist> 是最外层的标记，<rule> </rule> 标记描述一条规则，其中 <head> </head> 描述规则头，<option> </option> 描述规则选项。<rulelist> 和 </rulelist> 之间包含多个 <rule> </rule>，表示规则文件可由多条规则组成。这种规则结构结构清晰、直观易懂。

元素 head 和 option 中的 attribute - list 是定义的属性列表。

元素 head 包含的属性有：action，proto，sip，sport，dip，dport。以下是各个属性的含义的具体介绍。

(1) action：定义规则操作，就是在一个数据包满足所有在规则中指定的属性特征的情况下，所应该采取的行动。有以下四种属性值(四种动作)：alert 表示使用选定的告警方法警报，然后记录该数据包；log 表示记录该数据包；pass 表示丢弃该数据包；close 表示切断网络的物理连接。

(2) proto：定义协议项，说明规则定义的这种攻击应用的是何种协议。属性值有 TCP、UDP 和 ICMP 等。

(3) sip：定义数据包的源 IP 地址。

(4) sport：定义数据包的源端口。

(5) dip：定义数据包的目的 IP 地址。

(6) dport：定义数据包的目的端口。

"HOME_NET"代表内部网，"EXTERNAL_NET"代表外部网。"any"可以代表任意的 IP 地址或端口。网络数据包的流向统一指源地址 sip 流向 dip(sip - > dip)。

下面具体介绍元素 option 包含的各个属性的含义。

(1) msg：设置具体告警信息。

(2) logto_filename：将数据包记录到一个用户指定的文件中。

(3) ttl：测试 IP 数据包的 TTL 域。

(4) id：检测 IP 数据包的分段 ID 域是否等于指定的值。

(5) dsize：检测数据包的有效荷载是否等于指定的值。

(6) content：在数据包的有效荷载中搜索指定的模式串。

(7) offset：设定 conent 中所指定的起点。

(8) depth：设定 conent 中所指定的终点。

(9) nocase：设定搜索中使用与大小写无关的方式。

(10) flags：测试各种 TCP 标识值。

(11) seq：检测 TCP 的顺序号是否等于指定的值。

(12) ack：检测 TCP 的应答域否等于指定的值。

(13) itype：检测 ICMP 类型域是否等于指定的值

一条规则中，规则头元素 head 的属性必须是完整的，而规则选项 option 元素的属性是可选的。下面是一条规则的实例。

```
<rulelist>
<rule>
  <head action="log" proto="tcp" sip="EXTERNAL_NET"
  sport="any" dip="HOME_NET" dport="139"></head>
  <option msg="Log TCP packets"></option>
</rule>
</rulelist>
```

此条规则所表示的含义是：系统记录(log)从外部网络流向内部网络的端口为 139 的使用 TCP 协议的数据包。

若要检测蠕虫的攻击，可制定如下规则：

```
<rule>
  <head action="alert" proto="tcp" sip="EXTERNAL_NET"
  sport="any" dip="HOME_NET" dport="135"></head>
  <option msg="W32.Nachi.Worm virus"></option>
</rule>
```

4.2 规则的解析

由于规则文件是 XML 文档，所以我们可以使用 DOM 来解析规则文件。DOM(Document Object Model，文档对象模型)是一个对象化了的 XML 数据接口。DOM 定义了文档的逻辑结构，提供了对 XML 文档进行访问和操作的方法。利用 DOM 程序开发人员可以动态地创建文档，遍历文档结构，添加、删除和修改文档内容。利用 DOM 接口可以很方便地解析规则文件。系统将规则分解为链表头和链表选项进行引用。链表头由诸如源/目标 IP 地址及端口号这些普通信息标识。链表选项定义一些更详细的信息如 TCP 标志、ICMP 代码类型、特定的内容类型、负载容量等。系统的规则解析流程其实很简单：系统首先读取规则文件 XML 文档，紧接着依次读到每一条规则(标记 <rule>、</rule> 之间就是一条规则)；然后对其进行解析，在逻辑上组织成规则链。规则链按规则头中不同的协议类型进行分类组织。

使用 DOM 解析规则文件的基本过程如下：

(1) 获取 XML 规则文件的文件(Document)结点；

(2) 获取文件(Document)结点的所有子结点；

(3) 找出文件(Document)结点所有子结点中的 rulelist 结点；

(4) 获取 rulelist 结点的所有子结点；

(5) 取 rulelist 结点的一子结点；

(6) 判断此子结点是否为 rule 结点？是则继续，若不是转步骤(5)，直至取完 rulselist 结点的所有子结点；

(7) 获取 rule 结点的所有子结点；

(8) 取 rule 结点的一子结点；

(9) 判断此子结点是否为 head 结点？是则解析规则头，若不是继续；

(10) 判断此子结点是否为 option 结点？是则解析规则选项，若不是转步骤(8)，直至取完 rules 结点的所有子结点。

其中，规则头和规则选项的解析类似，先获取 head 结点(或 option 结点)的所有属性，然后将属性的值添加到相应的规则链中。

4.3 检测流程

检测流程实质上就是一个规则匹配的过程。检测引擎按照规则文件中定义的规则依次分析每个数据包，与数据包中数据匹配的第一条规则触发规则定义中指定的动作。规则匹配就是对从网络上捕获的每一

(下转第 47 页)

(上接第 43 页)

条数据报文和规则解析后形成的规则链进行匹配,如果发现存在一条规则匹配这个报文,就表示检测到一个攻击,然后按照规则指定的行为进行处理(如切断物理连接、发送警告等);如果搜索完所有的规则都没有找到匹配的,就表示报文是正常的报文,没有发生入侵事件。

5 结束语

入侵检测系统作为一种主动安全防御系统,逐渐发展成为保障网络系统安全的重要部件。从入侵检测技术出现至今,产生了多种不同类型的入侵检测系统。怎么样实现各种 IDS 之间的互操作,是 IDS 标准化工作研究的一个重要目标。提出用 XML 来描述规则,可实现入侵检测系统各部件之间所交换数据形式的标准化,是 IDS 系统实现互为通用,不同入侵检测系统之间实现互操作成为可能的一种途径。

参考文献

1 系统安全与入侵检测,戴英侠等,北京 清华大学出版社,2002。

2 Thomas H. Ptacek and Timothy N. Newshan. "Insertion, Evasion, And Denial Of Service: Eluding Network Intrusion Detection", Technical Report, Secure Networks, Inc. January 1998.

3 W. Lee S. J. Stolfo, and K. Mok," A Data Mining Framework for Building Intrusion Detection Models", Proceedings of the IEEE Symposium on Security and Privacy, 1999.

4 CIDF specification documents." The Common Intrusion Detection Framework Architecture".