基于 Agent 技术的多源日志采集系统的设计与实现

Design and Implementation of Multi – source Log Collect System Based on Agent Technology

刘必雄　（福建农林大学计算机与信息学院 福建福州 350002）
（福州大学数学与计算机学院 福建福州 350002）
魏　连　（福州大学数学与计算机学院 福建福州 350002）
许榕生　（中国科学院高能物理研究所计算中心 北京 100049）

摘　要： 在分析 Agent 技术以及 4 种日志采集技术的基础上，本文提出了一个基于 Agent 技术的多源日志采集系统，来实现采集各种类型的系统日志。本文详细分析了系统的基本架构以及日志采集 Agent 的结构模型和工作流程。最后探讨了基于正则表达式的日志数据的识别与抽取以及日志 XML 格式化的实现技术。

关键词： Agent 技术　日志采集技术　正则表达式　XML

1　引言

日志审计系统通过对网络系统中各类的日志进行审计分析，来协助安全分析人员及时发现网络薄弱环节，采取有效措施，提高安全等级，具有重要意义。要构建一个多源日志集中审计系统首先应当将分布在网络各个角落的设备和系统日志数据集中起来，才能进行统一管理和分析，因此必须考虑如何采集网络环境下的多源日志数据的问题。文献[1]通过修改 Linux 内核来转移审计日志到运行在另一个虚拟机上审计后台进程来实现日志审计分析功能；文献[2]通过监控代理对局域网的监控日志和终端日志进行收集以及审计分析；文献[3]给出了一个多源日志审计系统的设计方案。这些文献着重研究日志审计技术，而对日志采集问题并没有进行详细地说明。本文设计并实现一个日志采集系统，它能够运行在网络中各个日志采集点上，负责监控和采集新生成的各种日志，并及时地将日志数据发送到日志服务器。由于 Agent 具有自主性、协作性以及反应性等智能特性，能够自主地运行于复杂环境中，完成给定的任务，因此采用 Agent 技术来实现，可以解决网络中各种类型日志采集的问题。

2　相关技术

2.1　Agent 技术

Agent 是人工智能和计算机软件领域的一种新兴技术，现广泛地应用于人工智能、网络管理、软件工程等领域[4]。有关 Agent 的定义，至今为止学术界还没有一个统一的、确定的定义。软件 Agent 研究者一般认为软件 Agent 是指能在特定的环境下无须人工干预和监督而自主完成某项任务的计算实体，并能与其它 Agent 进行必要的交互。Agent 具有自治性、反应性、协作性以及适应性等显著特点。

2.2　日志采集技术

日志采集技术可以分为主动（Active）采集和被动（Passive）采集两种模式。主动采集模式主要是通过读取日志文件来实现日志采集；被动采集模式是通过 Syslog、SNMP 以及 OPSEC 等协议来实现日志采集。

（1）基于文件读取的日志采集技术：该技术针对原始日志数据以文件的形存储在一个固定的位置（如

Web 日志等)的情况,可以直接对日志文件进行读取来获取日志数据。

(2) 基于 Syslog 协议的日志采集技术:Syslog[5]提供了一种传输方式,使机器能够通过 IP 网络传送事件的通告信息到 Syslog 服务器。通过配置网络设备,将日志数据以 Syslog 协议方式发送到指定的 Syslog 服务器。

(3) 基于 SNMP 协议的日志采集技术:SNMP 提供了一种从网络设备中收集网络管理信息的方法。通过对支持 SNMP 协议的网络设备进行配置,可以在设备中侦听 UDP 端口(161 和 162),取得特定的日志数据,然后将日志数据传送到日志服务器。

(4) 基于 OPSEC 协议的日志采集技术:OPSEC Software Development Kit(SDK)是由 OPSEC LEA 提供了,它定义的采集日志的接口,并且将所有网络通讯的具体实现全部封装[3],可以利用 SDK 采集支持 OPSEC 协议的防火墙以及 VPN 设备中的日志数据。

3 多源日志采集系统的设计

3.1 系统基本架构

日志采集系统是部署在网络中要监控的各个节点上,包括服务器、主机、安全设备以及网络设备,负责调度各个日志采集模块,实现日志采集、日志处理和日志发送的功能。日志采集系统的基本架构如图 1 所示:

图 1　日志采集系统的基本架构

日志采集系统能够采集操作系统、应用系统、安全设备以及网络设备所产生的各种类型的日志数据,并将采集到的日志数据发送给日志服务器。目前国际上日志还没有统一标准,不同厂商的设备或系统所产生的日志在语法和语义都存在很大的差异,其产生方式和存储方式也都有各自的特点,显然单一的采集技术很难实现采集多源日志数据。因此,本系统采用文件读取方式、Syslog 协议方式、SNMP 协议方式以及 OPSEC 协议方式等 4 中采集技术相结合的方式,来实现采集各种类型的日志。

日志采集系统的关键部件是轻量型日志采集 Agent,它分布于整个网络中各个采集点上,实现对日志数据采集、处理、按统一的 XML 格式对日志数据进行转化,最后将格式化后日志文件发送给日志服务器。它可以由多个 Agent 协同操作来共同完成日志采集工作,各个 Agent 之间的通信是以黑板为媒介,进行交流、协调任务进度和防卫,从而可以避免重复操作和死锁。

日志采集 Agent 与日志服务器之间的通信使用 TCP 协议,并通过 SSL(Security Socket Layer,安全套接层)协议进行加密和认证。由于日志文件中包含了非常重要、敏感的信息,采用 SSL 加密机制可以防止日志信息被窃听。另外,为了避免任何主机都可以向日志服务器发起 TCP 连接并发送虚假的日志文件,使用 SSL 的公证机制可以防止日志服务器接收到虚假的日志文件,保证只有经过认证的日志采集 Agent 发送的日志文件才会被接收,从而保证日志文件的可信性和可靠性。

3.2 LCA(Log Collect Agent)的结构模型

Agent 结构是构造和实现 Agent 功能的基础,是研究如何将表示 Agent 特性的模块,关联成一个有机的整体,来更好地实现对输入信息有效地处理,使 Agent 产生合理的动作来影响环境和 Agent 自身将来的状态。在日志采集 Agent(Log Collect Agent,简称 LCA)中主要包括日志采集模块、日志处理模块、日志发送模块、采集间隔时间计算模块、时间同步模块以及通信模块,其结构模型如图 2 所示:

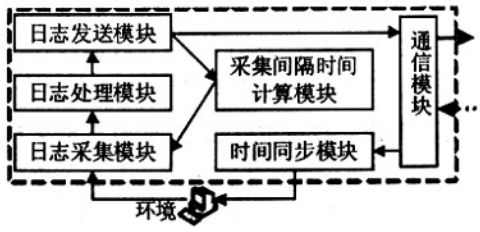

图 2　日志采集 Agent 结构模型

LCA 的各个模块的功能如下:

(1) 日志采集模块:根据采集间隔时间,定期采集系统日志;

(2) 日志处理模块:对采集到的日志进行 XML 格式化,生成新的日志文件;

(3) 日志发送模块:将日志文件发送到日志服务器;

(4) 采集间隔时间计算模块:根据系统日志记录数量的变化,对未来系统日志产生的日志记录数量进行预测,动态地调整采集间隔时间以适应系统实际需要;

(5) 时间同步模块:根据 NTP 协议保证采集点的系统时间与日志服务器时间同步;

(6) 通信模块:各个 LCA 之间的通信以及 LCA 与日志服务器之间的通信。

LCA 具有 Agent 的基本特性,有以下几个特点:

(1) 自治性:LCA 被部署到采集节点后,便可以自动在采集节点中运行并采集相关的日志,而无须人工干预。

(2) 反应性:LCA 可以检测所在的采集点上的状态,并根据系统的状态变化采用相应的动作,比如当检测到采集点的系统时间与时间服务器的时间不一致时,LCA 就会重设采集点的系统时间使之与时间服务器同步。

(3) 适应性:LCA 可以根据采集点的日志记录数量的变化,来调整自身的运行状态,如在某段时间内采集点产生的日志数据比较多的,Agent 就会调整采集的时间间隔,以避免每次采集量过多。

(4) 协作性:LCA 之间可以相互协作完成日志采集任务,LCA 还可以与其它 Agent(如日志接收 Agent)相互协作完成日志数据发送任务。

3.3 LCA 的工作流程

日志采集 Agent 部署在网络中各个采集点上,实现收集各种类型的日志数据,其工作流程如图 3 所示,主要由采集过程、处理过程以及发送过程三个过程组成。

4 关键模块的实现

4.1 基于正则表达式的日志识别与抽取

正则表达式[6](Regular Expression)一种基于模式匹配和替换的强有力的字符串分析工具,它提供功能强大、灵活而又高效的方法来处理文本,它通过全面模式匹配可以快速分析大量文本找到特定的模式,从而

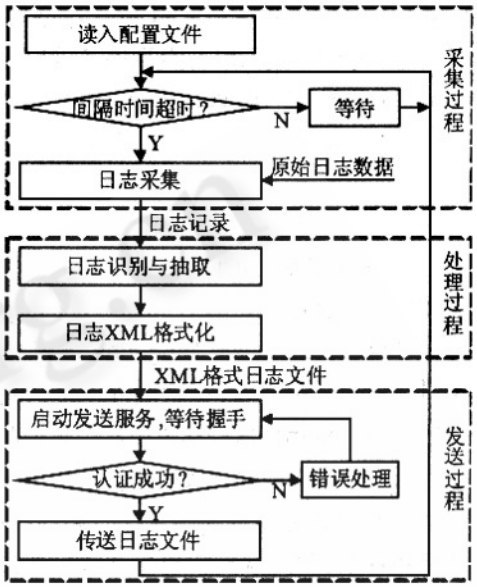

图 3　日志采集 Agent 的工作流程

可以根据模式匹配从字符串中提取一子字符串。采用正则表达式来实现,可以简化日志内容的识别的难度,减少程序中可能存在的错误;另外,采用正则表达式还可以提高程序的灵活性与通用性。因此,本文采用了正则表达式来实现对日志数据的识别和有效数据的抽取。

由于每种类型的日志的表示格式是固定,所以可以根据日志格式和需要抽取的字段,为日志格式定义一个正则表达式。比如 IIS 的 Web 日志记录:"2007 - 07 - 19 01:20:40 W3SVC1 192.168.10.33 GET /svn - 80 - 192.168.10.70 301",每个日志字段之间以一个空格分开,依次表示"date"、"time"、"s - sitename"、"s - ip"、"cs - method"、"cs - uri - stem"、"cs - uri - query"、"s - port"、"cs - username"、"c - ip"以及"sc - status"。其抽取相关日志字段的正则表达式如下所示:

(? <vdate>\d{4} - \d{2} - \d{2}) 获取日期

\s{1}(? <vtime>\d{2}:\d{2}:\d{2}) 获取时间

\s{1}\ (\S{1,100}) 忽略 s - sitename

\s{1} (? <sip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})

获取服务端 IP 地址

\s{1} (? <vmethod>GET|POST|OPTIONS|

HEADER|PUT) 获取访问请求方式

\s{1}\(\S{1,100}\) 忽略 cs-uri-stem 字段

\{1}\(\{1,100}\) 忽略 cs-uri-query 字段

\{1}(? <sport>\{1,5}) 获取服务端口号

\{1}\(\{1,100}\) 忽略 ccs-username 字段

\s{1}(? <cip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})

获取客户端 IP 地址

\s{1}(? <vstatus>\d{3}) 获取状态码

4.2 基于 XML 日志格式化

XML[7](Extensible Markup Language,可扩展标记语言),由于具有独立于平台、可扩展性以及自描述性等特性,已经成为 Internet 异构环境中不同类型和不同领域数据表示和数据交换的开放标准。它能够使各种不同来源的结构化数据很容易结合在一起,以统一的格式来表示各种数据源,从而实现各种数据集成。本文将日志记录转化为 XML 格式,其格式如下:

```
<record>
  <date> </date>
  <time> </time>
  ...
</record>
```

其中 <record> 和 </record> 标志对表示一条日志记录,<date> 和 </date> 标志对之间的表示日志事件发生的日期,<time> 和 </time> 标志对之间的表示日志事件发生的时间等。

在日志处理模块中,需要将本次采集的所有日志数据转化为 XML 日志文件。在日志数据识别与抽取过程中,已经将原始日志记录中的字段内容存储在结构体的数据项中(如 vdate 存放日期字段内容,vtime 存放时间字段内容),这样只要根据结构体的数据项与 XML 格式的元素的对应关系,就可以实现转化过程,其主要算法如下:

```
创建一个输出文件;
将固定的 XML 格式的文件头写入输出文件;
While(有采集的原始日志数据输入){
  取得日志记录中的数据字段;
  For(每一条日志记录){
    取得对应字段的数据;
    按照 XML 格式写入数据文件;
  }
}
关闭输出文件;
```

将采集到的日志数据转化为一个标准的 XML 文件,在该文件中不但包括所需的日志记录字段,而且能更清晰地表达出每个字段数据的确切含义,为日志数据解析、分析以及统计提供了有利的条件。

5 结束语

本文采用 Agent 技术设计并实现了一个多源日志采集系统,运行在各个采集点上,实现收集分散于网络各个节点中的日志数据。本文在详细分析 Agent 技术以及各种类型的日志采集技术的基础上,给出了多源日志采集系统的基本架构,并论述了日志采集 Agent 的结构模型以及工作流程。最后深入探讨了日志数据的识别、抽取以及格式化的实现技术。

参考文献

1 孟江涛、卢显良、聂小文,一个基于虚拟机的日志审计和分析系统[J],计算机应用,2006,26(12):2913-2915,2918.

2 杨艳、刘育楠,一种基于局域网络监控日志的安全审计系统[J],计算机应用,2007,27(2):292-294,298.

3 黄艺海,日志安全审计系统的设计与实现[D],浙江:浙江大学,2006.

4 Wooldridge M. J and N. R. Jennings. Intelligent Agents: Theory and pratice[J]. The Knoledge Engineering Review,10(2):115-152, 1995.

5 Karen Kent. Guide to Computer Security Log Management. [EB/OL]. (2006-5-1)[2007-8-12]. http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf.

6 杨桢、赵燕平、朱东华,基于正则表达式的信息抽取系统在国防技术监测中的应用[J],北京理工大学学报,006, 26(z1):74-78.

7 Extensible Markup Language (XML)[EB/OL]. (2006-09-26)[2007-04-28]. http://www.w3.org/XML/.