流量分析系统 TAS 的设计实现*

沈华林, 张　辉, 徐同阁
(北京航空航天大学 软件开发环境国家重点实验室, 北京 100083)

摘　要: 流量分析系统可以应用于网管,也可以作为入侵检测系统的一部分。提出了一种可扩展的流量分析系统,可以使用各种插件来处理报文,以统计流量和检测入侵。PCAP 采集端捕获报文后,交处理端分析。该处理端使用加权流量对主机列表进行排序,根据二八原则来减少内存的消耗。然后利用各种插件对报文、处理端结果进一步进行分析。在插件部分,介绍了插件的结构和数据导出插件的原理,该导出插件基于 RRD。

关键词: 流量分析; 插件; 加权流量; 二八原则; RRD
中图法分类号: TP393.07　　文献标识码: A　　文章编号: 1001-3695(2006)03-0251-03

Design and Implementation of Traffic Analysis System

SHEN Hua-lin, ZHANG Hui, XU Tong-ge
(National Key Liboratory of Software Development Environment, Beihang University, Beijing 100083, China)

Abstract: Traffic analysis system is used not only for network management but also for intrusion detection. An extensible traffic analysis system based on plugin structure is designed. Customers can select and develop their own plugins to manage network and detect intrusion. Collector module uses Packet Capture(PCAP) library to capture packets, then disposal module uses weight traffic and 20% ~80% principle to reduce memeory consume. Output plugin is based on RRD.

Key words: Traffic Analysis; Plugin; Weight Traffic; 20% ~80% Principle; RRD(Round Robin Database)

现有的流量统计工具和入侵检测系统,其采集端大多基于 PCAP 库。系统在抓包后再进行分析处理。Ntop[1,2], Snort[3] 和 MENet[4] 都是基于这种原理。流量统计工具主要通过分析报文头,统计得出被监控网络的总体信息;入侵检测系统则更多的关注于报文内容,通过模式匹配发现攻击特征,如使用 Bloom Filters[5] 数据结构进行快速的匹配定位。实际上,很多流量统计工具和入侵检测系统都不是纯粹的单一功能系统,它们同时还具备其他作用。协议/流量分析工具 Ntop 不仅可以通过分析报文头得到网络中的协议分布,还可以通过分析 FTP 报文内容以记录匿名的 FTP 服务器列表;入侵检测系统 Snort 不仅可以检测网络入侵还可以作为 Tcpdump 的替代工具来存储和显示报文。所以,在设计流量分析系统时,需要考虑其可扩展性,既能为网管服务也可用于入侵检测。本文中提出了一种插件式的流量分析系统,该系统能够捕获网络报文并进行简单的处理,同时提供了一种插件式体系结构,利用这些特定的插件可以完成特定的功能。用户可以根据自己的需求来选择其中的几种插件,也可以扩展开发自己的插件。

该流量分析系统(Traffic Analysis System)在下文表述中均用 TAS 来指代。

1　系统框架

TAS 的体系结构如图 1 所示,系统由数据采集端、加权流量处理端和插件三个部分组成,数据采集端基于 PCAP 库开发而成,将采集网卡设置为混杂模式后,从被监控网络捕获所需的报文,并将捕获的报文存储在报文队列中。加权流量处理端对队列中的报文依照二八原则进行处理,并将处理结果存放在主机 Hash 链表中。

而各种插件则对队列中的报文和主机 Hash 链表进行进一步的分析处理。

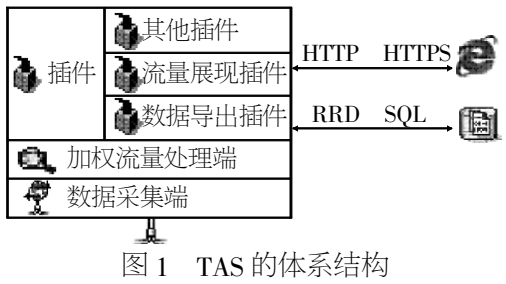

图 1　TAS 的体系结构

2　系统的设计实现

2.1　加权流量的设计思想

当数据采集端从网络中采集到报文信息后,将报文转交数据处理端进行统计分析。为了全面地展现整个网络中的流量分布,TAS 需要存储网络中所有主机的流量信息,该信息在内存中以主机 Hash 链表方式存在。当主机数目过多、监控的信息量过大时,往往会导致较大的内存消耗。

为了解决这个问题,Ntop 开发者 Luca Deri 使用了一种简易的方法。Luca 在 Ntop 中建立了一个特殊的线程,专门用于清除主机 Hash 链表中的不活动主机,每隔两分钟检查一次,如果存在某台主机半小时内没有任何流量,那么该主机的信息将被清除。另外,Ntop 中设置了一个全局变量用于监控主机数目的上限,一旦活动主机的总数目超过了设定的最大值,那么

收稿日期: 2005-01-14; 修返日期: 2005-04-13
基金项目: 国家“863”计划资助项目(2003AA1Z2240)

将不会处理新的主机信息。

Ntop 的做法存在很多不够合理的地方: ①不活动主机的界定是不科学的,假设某台主机每隔一小时产生大量的流量,而该流量都是在前半个小时内产生的,那么根据 Ntop 的规则,这台主机的流量会被定期地清除掉,这种清除的方式不合理; ②Ntop 中设置了监控主机数目的上限,那么超过上限的主机信息将不会被存储,如此,后采集的关键主机信息可能被丢弃。

所以必须采取一种新思路来解决这个问题。TAS 系统采用 Top *N* 加权流量来处理这个问题。由于网管人员最关心的是流量最大的 Top *N* 主机群的统计信息(*N* 可由用户指定) ,其他的主机信息则可以被忽略。而根据二八原则[6], 20% 的主机将产生 80% 的流量,所以 TAS 仅仅关注流量最大的前 20% 主机群的流量。

为了验证二八原则在流量监控中的正确性,笔者在不同环境下做了测试,实验结果如表 1 所示。

表 1 不同环境下流量二八原则的测试结果

单位	公司甲	公司乙	北航
主机总数	129	50	10 245
前 20% 主机数	26	10	2 049
主机总流量	253 MB	2.4GB	6.54GB
前 20% 主机流量	223 MB	1.83GB	6.17GB
百分比(%)	88.3	76.3	94.4

上述的三个实验分别是在公司甲的出口网关、公司乙的信息通信部门和北京航空航天大学网络中心出口网关处获取的流量信息。可以看到当主机数目较少的时候,基本符合二八原则。而当主机数目较多时,如在北航出口流量统计中,前 10% 的主机共计 1 024 台,总流量 5.58GB, 占总流量的 85.4% 。也就是说如果关注前 20% 的主机,将会基本满足试验的需求。

TAS 维持一个主机总流量的列表,将位于前 10% 流量的主机群列为第一集团,即用户最关心的主机群; 将位于 10% ～ 20% 的主机群列为第二集团,即有可能成为第一集团的主机群。第一集团和第二集团的主机群都会有详细的流量信息统计,包含网络层、传输层和应用层各种协议的具体信息以及总流量信息等统计。这两个主机群的信息分别存 储在两个不同的 Hash 主机链表中,其余主机的具体信息将不会被保存,而仅在总流量列表中保留概要信息。

在 TAS 中,用户对流量的关心程度会随时间而改变,越早的流量重要性越低。因此 TAS 的主机总流量列表中存放的不是实际的总流量,而是经过时间修正后的总流量。这种包含时间系数的流量,在 TAS 中称为加权流量。其中与时间相关的系数称为加权系数。每隔一小时, TAS 就将每台主机的加权流量 *T* 与加权系数 *A* 相乘得到新的加权流量 *T*′,即: $T' = T \times A$。

考虑到三天前的流量信息对用户意义不大,则设定三天前的加权流量为最初加权流量的 10% 。而 TAS 每小时处理一次,所以设定 *A* 的 72 次方为 0.1。*A* 求得为 0.968。

在 TAS 中,第一集团和第二集团中的主机往往会发生变化,所以必须依靠加权流量列表中的数据来进行更新。TAS 每隔固定时间对第一集团和第二集团的主机进行一次更新。

2.2 插件的原理

采用插件的目的在于扩展系统功能。在数据采集端采集到了所需的报文、加权流量处理端进行了统计分析以后,需要对报文内容进行进一步分析处理,以及存储加权流量处理端的处理结果。这些不同的功能可以使用不同的插件来完成。

在 TAS 中,插件的核心数据结构如下:

```
typedef struct pluginInfo {
  char * pluginName; //插件名称
  char * pluginDescr; //插件的详细介绍
  char * pluginAuthor;
  char activeByDefault; //默认状态是否激活
  IntFunc initFunc; //启动插件所调用的函数
  VoidFunc endFunc; //结束插件所调用的函数
  char * pluginStatusMessage; //插件状态信息
} PluginInfo;
```

在该结构中定义了启动插件、结束插件所调用的函数接口。其中, initFunc 用于插件的初始化, endFunc 用于结束插件。具体定义如下:

```
typedef void( * VoidFunct) ( void) ;
typedef int( * IntFunct) ( void) ;
```

这些已经设计好的插件都以动态链接库的方式存放于固定目录下,文件名和插件的名字保持一致,方便系统调用。

TAS 加载插件时,从插件目录中读取文件名,获得指定插件的调用接口:

```
PluginInfo* PluginEntryFunc( ) {
   return( pluginInfo) ;
}
```

利用该接口函数获取插件的入口指针 pluginInfo,进而执行该插件的初始化函数 initFunc,然后启动一个专门的线程完成插件的功能。

插件的处理对象是程序中的全局变量,也就是报文队列以及主机 Hash 链表。

卸载插件时调用 endFunc 结束 initFunc 所启动的线程。

2.3 数据导出插件

TAS 启动数据导出插件后,会建立一个单独的线程存储分析结果,数据库是基于 RRD。RRD(Round Robin Database) 它是一种特殊的大小不会增长的数据库,专门记录某段时间内的一组数值。

RRDTOOL[7,8] 是操作 RRD 的工具,它包含了 RRD 函数访问接口。TAS 通过集成 RRDTOOL 的相关代码,实现了对 RRD 的直接访问。

RRD 由一组 RRA(Round Robin Archive) 构成。每个 RRA 都可以看作一个循环链表,在这个链表中,记录了最近一段时间的数据。当 RRA 已满,最新数据来到时,就用该数据替代最原始的数据,从而保证了 RRA 的总条目数不会发生改变。由此可以看出, RRA 最大的特点就是大小恒定。

为了记录时间更加久远的流量数据, RRD 中的各个 RRA 采集的时间粒度各不相同,以确保单个 RRD 中能够记录足够长时间内的历史数据。在 RRD 中记录的数据,时间相隔越远,粒度越粗,时间相隔越近,粒度越细。

TAS 中每个 RRD 都由三个 RRA 构成,第一个 RRA 的默认时间间隔是 5min,总共拥有 864 条记录,也就是记录了三天内的流量数据,用以生成日流量报表。这个默认时间间隔是可以改变的,以便用户可以记录得更加详细。在更改默认时间间隔的同时,记录的总数也发生相应的变化,以保证第一个 RRA 记录的是三天的流量数据。第二个 RRA 的默认时间间隔是一

小时,总共拥有 2 160 条记录,也就是记录了 90 天大约三个月的流量数据,用于生成周流量报表和月流量报表。第三个 RRA 的默认时间间隔是一天,总共拥有 1 080 条记录,也就是记录了 1 080 天大约三年的流量数据,用于生成年流量报表。采用这种记录方式,TAS 可以记录大约三年内的历史流量数据。但由于单个 RRD 文件只能够存放单台主机某特定协议(如 TCP)的流量分布,所以必须有一个机制用来存放所有协议的 RRD 文件。最简单实用的方式是采用基于 RRD 的文件系统。图 2 是该文件系统的具体结构图。

图 2　TAS 中 RRD 文件的存储结构

在指定 RRD 目录下,根据不同的采集网卡分为 eth0, eth1 等,而在指定的 eth 下面,具体分为以 IP 划分的外网主机(目录结构类似 202/112/131/11)和以 MAC 划分的内网主机(目录结构类似 00/50/BA/30/FE/74)这两大类。每一个顶层的目录都代表存储的一台主机,每个目录中存放了代表该主机各种协议和各种统计的 RRD 数据库文件。

采用 MAC 地址形式的目录结构来存储内网的流量,是为了保证内网主机流量的存储不会因为主机 IP 地址的变化而发生多次存储的现象。而外网主机则因为无法获取对方的 MAC 地址,所以不得不采用 IP 形式。

3　结束语

本文提出了一种开放式的流量分析系统,使用该系统,用户可以完成流量统计功能,并能开发自定义插件以实现入侵检测。系统进一步的开发目标是开发出各种入侵检测插件,以扩大系统的使用范围。

参考文献:

[1] uca Deri. Monitoring Networks Using Ntop[C]. Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on, 2001. 199-212.

[2] Luca Deri. Effective Traffic Measurement Using Ntop[J]. Communications Magazine, IEEE, 2000, 38(5): 138-143.

[3] Brian Casw, *et al*. Snort 2.0 入侵检测[M]. 宋劲松,等. 北京:国防工业出版社, 2004.

[4] Naveed Ahmed. TCP/IP Protocol Stack Analysis Using MENet[C]. ENCON 2003 Conference on Convergent Technologies for Asia-Pacific Region, 2003. 1329 - 1333.

[5] Sarang Dharmapurikar, *et al*. Deep Packet Inspection Using Parallel Bloom Filters[C]. Proceeding of the 11th Symposium on High Performance Interconnects, 2003.

[6] 刘亚. 论图书分库的二八原则[J]. 公安大学学报, 1999.

[7] Luca Deri. RRD and Ntop[EB/OL]. http://www.ntop.org, 2002.

[8] Tobi Oetiker. RRDTOOL[EB/OL]. http://people.ee.ethz.ch/~oetiker/Webtools/rrdtool/, 1999.

作者简介:

沈华林,男,江西景德镇人,硕士研究生,主要研究方向为计算机网络管理、计算机网络安全;张辉,男,副教授,主要研究方向为计算机网络;徐同阁,男,副教授,主要研究方向为计算机网络。

(上接第 233 页)

在软件工程领域,已经有较多的研究逐渐强调模型驱动而非数据驱动,同时,对自动生成用户界面行为特征方面也有所研究[5],虽然这些研究开始得都较早,但是目前应用不多。

4　相关项目介绍

本文介绍的 Schema 动态生成用户界面方案是清华大学图书馆数字资源编目系统的核心技术。在实际应用中,由于数字资源种类繁多,描述信息各不相同,针对每种资源,图书馆开发出一个 Schema 文档来规范其描述数据。在研究 Schema 驱动生成用户界面之前,需要程序员为每个 Schema 文档生成输入界面,并且要在 Server 端利用 JAXB 生成多个 Java 类,来处理用户输入的数据。这个过程不仅烦琐,并且难以应付用户需求的频繁变更。采用动态生成用户界面之后,这个问题有望得到缓解。

5　结束语

本文对利用 XML Schema 动态生成用户界面的技术难点进行了分析,并考察了各种生成方式和技术手段。在此基础上,选择了稳妥有效的转换工具和目标语言(Java 和 HTML),最后提出并实现了可行的设计方案。对于同类问题,提供了一种有效可行的解决方法。在今后的研究过程中,将完善各种 Schema 元素的转换规则,加强用户定制功能,使产生的页面更加灵活、专业。

参考文献:

[1] asayasu Ishikawa. XForms the Next Generation of Web Forms[EB/OL]. http://www.w3c.org/MarkUp/Forms/, 2004-08-31.

[2] Henry S Thompson, *et al*. XML Schema Part 1: Structures Second Edition [EB/OL]. http://www.w3.org/TR/xmlschema-1/, 2004-10-28.

[3] Patrick Garvey, Bill French. Generating User Interface from Composite Schemas[EB/OL]. http://www.idealliance.org/papers/dx_xml03/papers/03-03-04.pdf, 2003-12-07.

[4] Kent Fitch. Schema Driven User Interface Generation [EB/OL]. http://ausweb.scu.edu.au/aw02/papers/refereed/fitch/paper.html, 2002-07-06.

[5] Angel R Puerta, Henrik Eriksson, John H Gennari, *et al*. Beyond Data Models for Automated User Interface Generation[C]. Glasgow, Scotland, UK: Proceedings of the Conference on People and Computers IX, Cambridge University Press, 1994. 353 - 366.

作者简介:

彭世新(1977-),男,山东潍坊人,硕士研究生,主要研究方向为工作流系统、模型驱动架构;董丽,女,博士研究生,主要研究方向为数字图书馆体系结构和互操作技术、数字资源长期保存技术及标准。