第34卷 第21期 计 算 机 工 程 2008年11月
Vol.34 No.21 Computer Engineering November 2008
·安全技术· 文章编号：1000—3428(2008)21—0126—03 文献标识码：A 中图分类号：TP393.08

基于异构机群的高速网络入侵检测系统

杨 锋，钟 诚，尹梦晓
(广西大学计算机与电子信息学院，南宁 530004)

摘 要：结合异构机群系统，提出一种基于双向驱动的分流算法，将高速数据流分为多个子数据流，把子数据流交由异构机群系统中最合适的节点处理，实现基于异构机群的高速网络入侵检测系统。实验结果表明，该系统保证了某时间段内具有相同源或目的地址的所有数据包发向同一个后端IDS引擎进行检测，能在高速网络环境下保持高检测率，并有效解决负载均衡问题。

关键词：入侵检测；异构机群；负载均衡

High-speed Network Intrusion Detection System Based on Heterogeneous Cluster

YANG Feng, ZHONG Cheng, YIN Meng-xiao
(School of Computer and Electronics Information, Guangxi University, Nanning 530004)

【Abstract】 This paper presents a diffluent algorithm based on bidirectional drive by applying heterogeneous cluster computing systems. It divides high traffic into several data streams, sends them to the most suited node in heterogeneous cluster computing systems, and implements the high-speed network intrusion detection system based on heterogeneous cluster. Experimental results show that this system can ensure that all data packages which take on the same source or destination with period of time are sent to the same IDS analysis engines, keep high detection rate on high-speed network and resolve the load balancing problem effectively.

【Key words】 intrusion detection; heterogeneous cluster; load balancing

1 概述

如何实时获取并分析高速网络信息流，使网络入侵检测系统能适应高带宽和高负荷的高速网络环境，成为当今网络安全领域急须解决的技术难题[1]。目前相关研究机构提出了多种不同方法和技术，主要包括：(1)软件硬件化。其开发周期和成本较高，用它实现对复杂协议的分析难度较大。(2)高速匹配算法。使用一些高速匹配算法来提高匹配速度，比如用BM算法实现字符串匹配的优化。由于此类算法只能提高单条规则匹配的效率，无法提高整个规则集匹配的效率，因此当规则数线性增长时，其匹配效率将线性下降[2]。(3)分流流量。将捕获的高流量网络数据通过专用分流设备，根据分流策略分为多个低流量的数据流，并交由位于后端的IDS引擎处理。目前的分流技术存在如下2个问题：1)为了实现准确分析，要求保证某时间段内具有相同源地址或相同目的地址的所有数据报文发向同一个后端IDS引擎进行检测；2)必须有效解决均衡问题，保证后端的IDS不超载、不空闲。

各种具有不同计算能力、不同存储能力、不同处理器个数的计算资源(工作站、个人计算机等)通过高速网络互连，实现了一种新的计算平台——异构机群。异构机群系统利用工作站和个人计算机进行分布式并行处理，以较低成本完成大规模、复杂问题的计算处理，适用于处理高速网络中的大流量网络数据。

2 基于异构机群的高速网络入侵检测系统

2.1 异构机群

异构机群是由一组相独立的不同计算机组成的集合体，其中的各个节点通过高性能互连网络连接，各个节点可以协同工作并表现为一个单一的、集中的计算资源供并行计算任务使用。近年来，适用于异构机群上的开发工具日趋成熟，异构机群系统正迅速发展[3]。

2.2 基于异构机群的高速网络入侵检测系统结构

在高速网络环境下，若只采用一台普通计算机作为检测器，必然会出现丢包现象。为了适应高速网络数据流的入侵检测，本文将捕获的高流量网络数据通过分流算法分为多个数据流，交由位于异构机群系统节点中的IDS分析引擎处理。此系统主要由数据采集器、机群系统节点中的IDS分析引擎、管理控制台等部分组成，如图1所示。

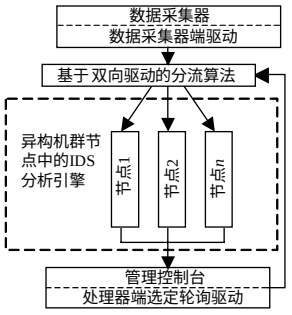

图1 基于异构机群的高速网络入侵检测系统结构

基金项目：广西自然科学基金资助项目(桂科自0339008)；广西大学科研基金资助项目(X071105)；广西高校人才小高地建设创新团队计划基金资助项目(桂教人[2007]71号)

作者简介：杨 锋(1979－)，男，讲师、硕士，主研方向：网络信息安全，并行计算；钟 诚，教授、博士；尹梦晓，讲师、硕士

收稿日期：2008-02-12 **E-mail：**yf@gxu.edu.cn

图 1 中主要组成部分的功能如下：(1)数据采集器。对高速宽带网段上的数据包进行高速采集，并根据本文提出的基于双向驱动的分流算法将高速的网络数据流分发到各节点。(2)异构机群节点中的 IDS 分析引擎。由多个进行入侵检测分析的处理器组成，在图 1 中，该系统共有 n 个处理器。其主要功能包括：1)各个节点对数据采集器传送过来的数据包进行分析，判断是否有入侵攻击或可疑行为；2)将检测结果保存并发往管理控制台进行综合处理，指导系统做出合适的响应动作；3)收集各个节点的待处理数据量、可用 CPU 和可用内存等信息，并发给管理控制台。(3)管理控制台。其主要功能包括：1)汇总各个处理器的分析结果，进行统一的协同处理，产生并显示报警和日志信息；2)生成处理器端选定轮询驱动信息，用于数据采集器执行基于双向驱动的分流算法。

3 基于双向驱动的分流算法

为了保证某时间段内具有相同源地址或相同目的地址的所有数据报文发向同一个后端 IDS 引擎进行检测，并保证异构机群系统中各个 IDS 分析引擎不超载、不空闲，本系统引入了基于双向驱动的分流算法。

3.1 数据采集器端驱动

只发送一两个数据包通常难以发现网络攻击，必须通过频繁发送大量数据包才能实现。因此，一段时间内的网络流量特征有助于检测攻击行为，可以通过设置时间窗来构造流量特征。参照文献[4]采用的方法，设置时间窗口为 2 s，可以构造如下特征：在过去 2 s 内与当前连接有相同目的主机的连接个数；在过去 2 s 内与当前连接有相同服务的连接个数。为了保证时间窗内具有相同源地址或相同目的地址的所有数据报文发向同一个后端 IDS 引擎进行检测，本文在数据采集器端驱动中构造了元特征组合函数。

定义 1(元特征组合函数) 元特征组合函数 $y(Src\_addr, Dst\_addr)$ 把与入侵特征密切相关的元特征 Src_addr 与 Dst_addr 映射到处理器标号空间$(1,2,\cdots,n)$的某个标号上。

$$y(Src\_addr,Dst\_addr)=h(Src\_addr,Dst\_addr,s) \bmod n,\ 1,2,\cdots,n \tag{1}$$

其中，Src_addr 和 Dst_addr 分别为源 IP 地址和目的 IP 地址；s 为时间窗，根据经验，本系统设 s=2 s；h 函数的作用是把字符串转换为整数。

通过元特征组合函数的映射关系，使在时间窗s时间段内具有相同源地址或目的地址的所有数据报文均映射到同一个标号上。系统在数据采集器中为每个处理器节点i建立相应的标记$flag_i$，并将所有$flag_i$赋初值为 0。如果在时间窗s时间段内已向处理器节点i转发过数据包，则$flag_i$=1，否则$flag_i$=0。在基于双向驱动的分流算法中，将根据$flag_i$的值确定到底是数据采集器端驱动还是处理器端选定轮询驱动。

3.2 处理器端选定轮询驱动

在异构机群中，各个节点具有不同计算能力和不同存储能力，某一时刻各个节点上的待处理的数据量不同。本文提出一种处理器端的驱动方法，根据异构机群中各个节点在某一时刻的待处理数据量、可用CPU和内存等因素，计算处理器的负载，采用选定轮询的策略，使异构机群系统中各个IDS分析引擎不超载、不空闲。为了量化异构机群中各个节点的运行情况，引入以下符号：(1)在t时刻第i台处理器的IDS分析引擎尚未处理的网络数据量表示为$D_i(t)$, i=1,2,⋯,n； (2)在t时刻处理器i的可用CPU表示为$C_i(t)$, i=1,2,⋯,n； (3)在t时刻处理器i的可用内存表示为$M_i(t)$, i=1,2,⋯,n。

定义 2(处理器负载函数) 处理器负载函数$f_i(t)$, i=1,2,⋯,n，其中，n为机群系统中处理器的总数，表示处理器i在t时刻的负载。综合待处理数据量、可用CPU和可用内存 3 个主要因素，可以得到处理器负载函数为

$$f_i(t)=D_i(t)/(a_iC_i(t)+b_iM_i(t)),i=1,2,\cdots,n \tag{2}$$

其中，a_i和b_i根据异构机群系统中节点i的计算能力和存储能力确定。

处理器端选定轮询驱动的处理过程步骤如下：(1)根据在数据采集器端驱动中设定的时间窗的值s，确定驱动时刻t，即每隔s时间段，驱动一次收集处理器端信息的动作。(2)计算t时刻的$D_i(t)$, $C_i(t)$和$M_i(t)$。(3)计算负载$f_i(t)$。(4)对步骤(3)计算出来的各个节点的当前负载$f_i(t)$进行排序。(5)根据步 骤(4)的排序结果，选出当前负载较小的前x个节点，并利用该x个节点的标号生成选定队列R，有 $1\le x\le n$，x根据实际网络流量和异构机群系统中各个节点的处理能力取经验值。(6)生成选定轮询驱动队列L，当驱动信号到来时，对输出驱动队列L中的最右端进行标号，并把队列L中的所有标号循环右移一位。上述处理过程如图 2 所示。

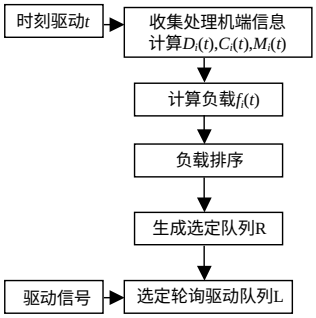

图 2 处理器端选定轮询驱动的流程

3.3 基于双向驱动的分流算法

基于双向驱动的分流算法如下：

输入：网络连接的数据报文 e。

输出：把输入的网络连接数据报文 e 发向异构机群系统节点中的某个节点。

算法步骤：

(1)利用元特征组合函数 $y(Src_addr,Dst_addr)$把数据报文 e 映射到处理器标号空间$(1,2,\cdots,n)$的某个标号 i。

(2)根据步骤(1)的i值读取相应的标记$flag_i$，如果$flag_i$=1，则采用数据采集器端驱动，直接把数据报文e转发给节点为i的处理器，转步骤(4)；如果$flag_i$=0，则给处理器端选定轮询驱动发出驱动信号。

(3)从选定轮询驱动队列 L 中获取接收数据报文 e 的处理器标号，并根据该标号转发数据报文 e。

(4)等待处理下一个数据包。

基于双向驱动的分流算法流程如图 3 所示。

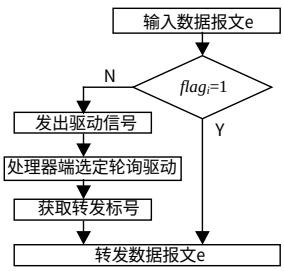

图 3 基于双向驱动的分流算法流程

此算法的数据采集器端驱动保证了时间窗在 s 时间段内具有相同源地址或相同目的地址的所有数据报文发向同一个后端 IDS 引擎进行检测。处理器端选定轮询驱动保证了异构机群系统中各个 IDS 分析引擎不超载、不空闲，有效实现了系统的负载均衡。

4 实验结果与分析

本文在共有 16 个节点的异构机群上进行实验，包括4台HP DX系列PC机(P4 3.0 GHz CPU, 512 MB内存)、2台IBM Netvista A系列的PC机(P4 2.0 GHz CPU， 256 MB内存)和10台IBM Netvista M系列PC机(P4 2.4 GHz CPU，其中，2 台内存大小为 1 GB，其余各台内存大小为 512 MB)，各个节点运行的操作系统为Red Hat Linux 9，各个节点上的IDS分析引擎为Snort。利用MIT林肯实验室提供的Kddcup99 数据集模拟高速网络流量[5]。

4.1 系统的检测率分析

本文通过网络流量和检测率之间的关系来分析算法的检测效果。当网络流量逐渐增加时，由于本文系统保证了时间窗在 s 时间段内具有相同源地址或相同目的地址的所有数据报文发向同一个后端 IDS 引擎进行检测，因此系统的检测率没有明显下降。例如在网络流量增加到 600 Mb/s 时，本文系统的检测率仍能保持在 0.95，而基于传统分流算法的系统的检测率已下降到 0.70。当网络流量增加到 1 000 Mb/s 时，本文系统能保持 0.92 的检测率，而基于传统分流算法的系统的检测率已快速下降到 0.45，如图 4 所示。实验结果表明，本文系统能在高速网络流量环境下保持高检测率。

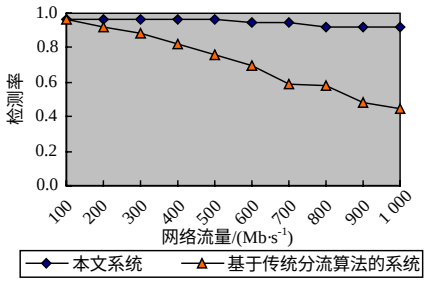

图 4 网络流量和检测率的关系

4.2 系统的负载均衡分析

本文通过网络流量和丢包率之间的关系来分析系统的负载均衡效果。在网络流量急速增加时，本文系统的丢包率没有急速上升，例如在网络流量增加到 600 Mb/s 时，丢包率仍维持在 0.03。当网络流量增加到 100 Mb/s 时，基于传统分流算法的系统的丢包率已达 0.03，当网络流量增加到 600 Mb/s 时，其丢包率急速上升到 0.20，如图 5 所示。实验结果表明，本文系统能有效地将捕获的高流量网络数据分为多个数据流，保证异构机群系统中多个 IDS 分析引擎不超载，有效解决了负载均衡问题。

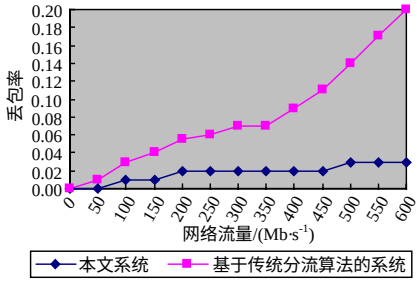

图 5 网络流量与丢包率的关系

5 结束语

本文把异构机群引入高速网络入侵检测系统中，提出基于双向驱动的分流算法。如果要更有效地实现高速网络环境下的入侵检测系统，除了解决网络数据的分流问题，还须在入侵检测分析模块中引入更高效的检测方法[6]，降低高速网络环境下入侵检测系统的误报率。

参考文献

[1] Song Bo, Ye Ming, Li Jie. Intrusion Detection Technology Research Based High-speed Network[C]//Proceedings of the 4th International Conference on Parallel and Distributed Computing. Chengdu, China: [s. n.], 2003.

[2] Jason C C, Staniford S, McAlerney J. Towards Faster String Matching for Intrusion Detection or Exceeding the Speed of Snort[C]//Proc. of DARPA Information Survivability Conference & Exposition. Washington D. C., USA: [s. n.], 2001.

[3] 陈国良, 安 虹, 陈 崚, 等. 并行算法实践[M]. 北京: 高等教育出版社, 2004.

[4] Li Wenke. A Data Mining Framework for Building Intrusion Detection Models[C]//Proc. of IEEE Symposium on Security and Privacy. Berkeley, California, USA: IEEE Press, 1999.

[5] Newman D. The UCI Knowledge Discovery in Databases Archive[Z]. (1999-10-28). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[6] 张克农, 陆佳华, 常羽飞. 基于 SVM 主动学习的入侵检测系统[J]. 计算机工程, 2007, 33(1): 153-156.

（上接第 125 页）

是利用程序中的数据流信息分析用户的输入信息是否会传播到访问敏感系统资源的代码处，如果答案是肯定的，则说明程序中存在特定的脆弱性。实验表明，依据此算法实现的软件能比较准确地判断被测 Java 程序中是否存在特定的脆弱性。

参考文献

[1] Dwibedi R. XPath Injection in XML Databases[Z]. (2005-07-01). http://palisade.plynt.com/issues/2005Jul/xpath-injection.

[2] 卜 莉, 李军怀, 张 璟. 基于 DTD 的 XML 与 SQL 查询转换算法[J]. 计算机工程, 2007, 33(16): 41-43.

[3] 戎 玫, 张广泉. 模型检测新技术研究[J]. 计算机科学, 2003, 30(5): 102-104.

[4] 汪小飞, 赵克佳, 田组伟. 数据流分析的关键技术研究[J]. 计算机科学, 2005, 32(12): 91-93.

[5] 刁树民, 王永利, 张晓勇. 一种数据流中奇异数据的自适应恢复方法[J]. 计算机工程, 2007, 33(15): 94-95.