

一种基于对等模型的网络入侵检测系统模型

李 兵

(中铁建电气化局集团第四工程有限公司, 湖南 长沙 410001)

摘 要:基于对等模型(Peer-to-Peer)的应用,提出一种分布式网络入侵检测系统:PeerIDS。该系统在设计上注重可靠性,且没有诸如单点失效一类的问题。入侵检测工作在由多台运行 PeerIDS 系统的连网计算机构成的对等网中随具体环境而自动进行迁移,以实现公平高效的分布式处理。同时,应用对等模型带来的可扩展性,使得该系统的性能可以通过简单地在网络中增加运行 PeerIDS 的计算机数目来不断提高,很好地适应了日益严峻的网络安全状况。在完成初始设置后,PeerIDS 系统的运行几乎不需要任何使用者的干预,体现了很好的自治性。

关键词:对等模型;网络入侵检测;分布式

中图分类号:TP393.08

文献标识码:A

文章编号:1673-629X(2008)03-0173-04

A Distributed Intrusion Detection System Based on Peer-to-Peer Model

LI Bing

(The 4th Engineering Co., Ltd. of China Railway Construction Electrification Bureau Group
Corporation, Changsha 410001, China)

Abstract: By employing the peer-to-peer(P2P)model, which is considered a promising approach to solve many problems in a distributed environment, presented a distributed network intrusion detection system named PeerIDS: an IDS solution values the properties of feasibility, durability and scalability most. Viewing the problem from a different perspective as against its counterparts, PeerIDS will provide the networked computation environment with robust and scalable protection while still stays efficient with the bumping of both types and traffic of malicious attacks through automatically and evenly distribute the intrusion detection workload among all the cooperating PeerIDS instances. Compared with many other distributed intrusion detection approaches, no single point of failure can be found in a form of synergized PeerIDS instances. Moreover, PeerIDS entails almost no additional administration work after the installation and first time setup.

Key words: peer-to-peer; net intrusion detection system; distributed system

0 引 言

由于在保护网络信息系统安全方面所起到的越来越重要的作用,入侵检测系统(IDS)近年来一直是一个研究热点。随着计算机网络的日益发展和网络规模的不断扩大,入侵方式亦日趋复杂,尤其是出现了一些相互协作的入侵行为,采用传统的集中式数据采集的传统入侵检测系统已经不能满足现实需要^[1]。于是,分布式入侵检测系统相应而生。但在传统的C/S网络体系结构下,如何有效地实现分布式入侵检测系统的数据分享,特别是分布式组件间的负载平衡和通信

问题,依然是未来亟需解决的问题^[2]。对等网(Peer to Peer, P2P)是一种新兴的网络通信模型,在对等计算、协同工作方面具有强大优势^[3],其应用正方兴未艾。

PeerIDS系统是一种分布式网络入侵检测系统,它试图把对等模型引入入侵检测领域。在由多台运行PeerIDS系统的连网计算机构成的对等网中,入侵检测工作随具体环境而自动进行迁移,以实现公平高效的分布式处理。同时对等模型的应用所带来的可扩展性使得该系统的性能可以通过简单地在网络中增加运行PeerIDS的计算机数目来不断提高,很好地适应了日益严峻的网络安全状况。

1 PeerIDS 系统简介

一个PeerIDS实例由三个主要构件组成:入侵检测引擎(IDS Engine)、状态检测模块(State Checker)、数据收发模块(Receiver/Transmitter, R/T)。其中IDS

收稿日期:2007-06-30

基金项目:国家自然科学基金资助(60673165);湖南省自然科学基金资助(05JJ30119);湖南省科技计划项目(2006JT1040)

作者简介:李 兵(1973-),男,湖南长沙人,工程师,研究方向为计算机通信安全。

Engine 负责执行具体的入侵检测功能。State Checker 将持续地监视其负荷状态并据之来决定入侵检测工作子集的装卸,再交由 Receiver/Transmitter 模块具体来执行与其他同伴间的入侵检测工作子集的收发以实现整个对等入侵检测网络的负载平衡。

与传统的入侵检测系统相比,PeerIDS 的入侵检测引擎因为通过分配至本地的入侵检测 Workload(互相正交的入侵检测工作的子集)来执行具体的入侵检测操作从而工作在一个更高的抽象级别上。不同于一些误用入侵检测方案中的规则集,PeerIDS 中的 Workload 抽象既包含入侵检测工作运行的工作数据集,如几条待执行的入侵检测规则或一组指定的待过滤的源 IP 地址掩码,也包含了在这些数据集上所需执行的操作,因而十分类似于面向对象方法学中对象(类实例)的概念。作为一种基于网络的入侵检测系统,PeerIDS 中所有被抽象为 Workload 的入侵检测工作子集均遵守一个以单个网络数据包为传入参数的共同接口。PeerIDS 系统可以通过其入侵检测引擎来集成检测不同类别攻击的 Workload 以实现一个功能完备的入侵检测系统。这些 Workload 的全体组成了整个入侵检测的问题空间。它们中的任何一员都可以通过 PeerIDS 同伴间的互相协作而运行在由多个 PeerIDS 实例构成的入侵检测对等网中的任何一个结点上。

状态检测模块是一个后台进程。它持续地监测入侵检测引擎的实时负荷状态并依据该状态执行相应的操作以实现所属 PeerIDS 实例的本地负载控制。如果入侵检测引擎占用的执行资源超出了某设定值,状态检测模块将把一条正存执行中的 Workload 暂停并挂入暂停工作集列表中,并将重复执行该操作直至入侵检测引擎的负荷恢复正常。类似地,当入侵检测引擎负荷低于某设定值时,为防止资源的浪费状态检测模块将逐条重启,直至入侵检测引擎具有正常负荷。

参与入侵检测对等网的各 PeerIDS 实例中的状态检测模块可以通过各自的数据收发模块进行一种间接的合作,从而实现了整个入侵检测对等网中工作量的负载平衡。数据收发模块在其所属 PeerIDS 实例的整个生命周期保持运行。当数据收发模块接收(定时阻塞以保持响应)到一条来自同伴的消息时,它首先判断该消息的发送者是否已存在于同伴列表中,如否,则在该列表中增加一条代表该消息发送者的条目。这样做也对 PeerIDS 系统被动的同伴发现机制作了很好的补充。随后,依据所收到消息的类型,数据收发模块将执行相应的数据收发及其所需的同步操作。在一轮循环执行完成之前数据收发模块将检查工作集列表,若其非空则将其首条 Workload 发送给同伴列表中当前

指针所指向的 PeerIDS 实例。

从整体上来说,在一个由多台运行 PeerIDS 的连网计算机(PeerIDS 实例)组成的对等网络内,正交的入侵检测工作子集(Workload)被自动分配到有足够执行能力的 PeerIDS 实例上执行。对等网中各 PeerIDS 实例上可用于执行入侵检测功能的资源(如 CPU 占用百分比)由用户指定或由各系统根据运行情况自行设定。有富余执行资源的 PeerIDS 实例可向其同伴请求更多的入侵检测工作。同样地,如果一个 PeerIDS 实例消耗完了指定的资源,它可以暂停其上运行的部分入侵检测功能,并尝试将其发送给对等网中的其他同伴以实现整个 PeerIDS 系统资源利用的最大化。当新的基于网络的入侵类型出现时,只需在对等网中运行着的任何 PeerIDS 实例上增加相应的入侵检测工作子集即可实现对该种入侵的检测,充分体现了整个系统的可扩展性。若网络流量及待检入侵种类的迅速增加而使得系统不堪重负的情况可以通过简单地向对等网中增加 PeerIDS 实例从而提高系统的容量来解决,提高了系统的可伸缩性。

PeerIDS 系统没有一个中央协调控制系统,成功地避免了单点失效问题,提高了系统的安全性。但是,网络攻击日益泛滥,尤其是出现一些互相协作的入侵行为^[1],PeerIDS 系统必须运行在对等模式下,这样,才能充分发挥由对等模型带来的可靠性、可扩展性及对等分布式系统在整体性能上(Capacity)的优势,成员发现和相互协作亦变得尤为重要。

2 成员发现及管理

为实现更好的自治性,PeerIDS 系统提供了同伴发现机制。为了建立一份处于活跃状态的同伴的列表,各 PeerIDS 实例在启动之初都会向局域中网络广播一条邀请消息,收到由其它同伴发出的回复消息后,彼此交换成员共享密钥^[4]。局域网中的 PeerIDS 实例依据各自收到的回答消息中的发送者信息建立起自己的同伴列表,从而在逻辑上构成了局域网的一个对等子网。在系统的正常运行中,如果一个 PeerIDS 实例发起通讯而某个同伴没有响应(或其在交互过程中由于某种原因失去响应),则它只需简单地把这个同伴标记为失活并试着与其他同伴完成相同的协作,而不必保持所有同伴的实时状态。这样做既简化了系统的设计也有效地减少了对网络带宽不必要的占用。各 PeerIDS 实例仅在启动时执行一次同伴发现操作。唯一的特例是当其需要与其他同伴通讯而同伴列表中的 PeerIDS 实例均被标记为失活,此时该 PeerIDS 实例将会再次执行同伴发现操作。

建立同伴列表后,如果没有负载平衡要求,各 PeerIDS 实例除了应答新加入实例的请求外,相互间不再通信。当整个入侵检测对等网中工作量进行负载平衡时,参与 PeerIDS 通过数据收发模块向同伴列表中的成员发送请求信息。接受者首先通过密钥对发送者是否已存在于同伴列表中进行判断,如否,则重新交换密钥,将其加进自己的同伴列表,这样做也对 PeerIDS 系统被动的同伴发现机制作了很好的补充。随后,将根据预设的协作机制,进行负载平衡的工作。

3 Pull 协作

当把全部本地入侵检测 Workload 投入执行而仍未充分利用给定的资源定量时,如图 1 所示,PeerIDS 实例(假设是 A)将向其 Peer List 中的某个同伴(假设是 B)发送一条 PULL 消息以请求承担更多的入侵检测工作量。收到请求后,B 将检查其处是否有挂起的 Workload,如没有则简单地返回一条 PULL REJECT 消息给 A。收到 B 的拒绝后,A 将向 Peer List 中的下一个同伴发送同样的请求消息,并将继续这一过程(以一定时间间隔以防阻塞)直至本地入侵检测引擎不再处于 HUNGRY 状态。

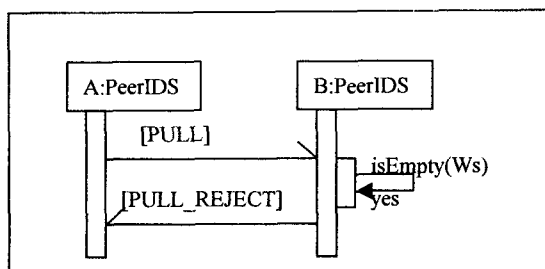


图 1 两个 PeerIDS 实例间的 PULL 协作(目标拒绝)

如果 B 中确有挂起的 Workload,则其将取出工作列表集中当前指针所指的 Workload 并将其装载入一条 Workload 消息发送至 A,如图 2 所示。在收到 B 发来的 Workload 消息后,A 将通过向 B 发送一条确认消息 DATA_ACK 来终止这次 PULL 协作。B 则在收到

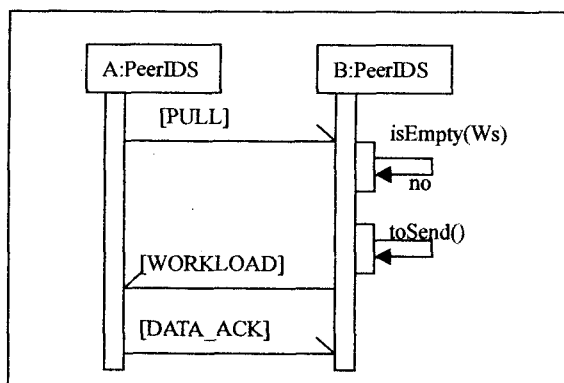


图 2 两个 PeerIDS 实例间的 PULL 协作(目标接受)

该确认后清除 SendTable 中与本次交互相关的信息并从工作列表集中删去已发送的 Workload。此时一部分入侵检测工作量就成功地从全负荷运行的 PeerIDS 实例 B 转移到相对空闲的 PeerIDS 实例 A。

4 Push 协作

当 PeerIDS 实例运行时所消耗的资源超过给定的范围时,状态检测模块将自动把一些入侵检测 Workload 从活动工作集列表移至等候工作列表集从而实现本地的负载控制。为了充分利用整个入侵检测对等网络的能力,具有挂起 Workload 的 PeerIDS 实例(假设为 A)将通过执行 PUSH 操作把这些 Workload 转发给其同伴(假设为 B)。收到 A 发来的 PUSH 消息后,B 首先检测本地入侵检测引擎的状态,若本地入侵检测引擎忙,则仅返回一条 PUSH_REJECT 消息给 A。收到拒绝消息后,A 将试着与 Peer List 中的下一个同伴建立 PUSH 协作。

若本地入侵检测引擎还有多余的执行能力,则 B 给 A 发送一条 PUSH_ACCEPT 消息以表示愿意接收更多的 Workload。A 将在收到该确认信息后把一条 Workload 发送给 B。在进行 PUSH 协作的过程中,信息的发送方 A 与接收方 B 分别通过在自己的 SendTable 和 RecvTable 中记录相应条目来跟踪此次交互。随着 B 返回一条 DATA_ACK 消息作为这次协作的结束,合作的双方各自删除与该次交互相关的跟踪信息,A 还要将已送出的 Workload 从工作集 Ws 中删去并重复该过程直至其 Ws 中的 Workload 全部发送出去或被状态检测模块恢复执行。

5 模型的实现

实现 PeerIDS 系统的最简单的方法是仅把它作为一种已有入侵检测系统的包装器(Wrapper)。在这种情况下,为了实现 PeerIDS 系统的本地负载平衡及在整个对等网中分布入侵检测的工作量,被包装的入侵检测系统必须具有可分的工作集。一种常见的可分入侵检测工作集即 Snort 系统的规则库。使各包过滤进程仅采样预先设定的特定类型的或处于特定范围内的数据包也可很好地做到对网络入侵检测工作的划分。在实现为包装器的 PeerIDS 系统中,Workload 将仅包含可分入侵检测工作集的一个子集而并不包含在这个子集上进行的操作。具体的入侵检测工作仍由被包装的入侵检测系统来完成。此时 PeerIDS 系统仅负责对等网络的构建及实现整个网络及其中各 PeerIDS 实例的负载平衡。以实现 Snort 的包装器为例,图 3 给出一种可能的 Workload 消息数据包。当收到这样一个消

息数据包后, PeerIDS 包装器实例将把附于其中的 Snort 规则添加到本地 Snort 系统的配置文件中(通常是 snort.conf, 在这里起到的是活动工作集列表 Wa 的作用)并回应发送者一条确认信息。随后再通过执行命令 snort -d -c snort.conf 重新启动本地 Snort 系统。若因为某种原因运行中的 Snort 进程占用了超出配额的资源, 则 PeerIDS 包装器将根据其超出的比例把 snort.conf 中的部分规则移至一个起到 Ws 的作用的本地文件如 suspended.conf 中并重启 Snort 系统以降低其负载。随后在执行 PUSH 操作时, PeerIDS 包装器实例把 suspended.conf 中的一条已挂起的 Snort 规则装载入一个 Workload 数据包发送至一个同伴并在收到该同伴的确认信息后删除该规则的本地拷贝。值得注意的是在入侵检测对等网中, 各 PeerIDS 实例所包装的 Snort 系统所执行的规则应相互正交而没有重叠。为做到这一点可以先清空各 Snort 安装的规则集, 当 PeerIDS 对等网建立起来后, 全部入侵检测规则可以通过任一 PeerIDS 实例注入对等网中并在其中自动分布。同样地, 为检测新种类的网络入侵也只需把相应的 Snort 规则添加至网络中任何一个 Snort 进程的配置文件 snort.conf 中, 系统的自动负载平衡功能将会把该规则移动到合适的 PeerIDS 实例上执行。

PeerIDS 系统运行所需的参数如: 状态检测模块两次操作间的时间间隔、收到拒绝消息直到再次发起同一操作的时间间隔、同伴间通信的超时设置等均和系统运行的网络及软硬件环境相关, 在 PeerIDS 的实现中这些值将是可配置的。对于文中多次提及的 PeerIDS 实例上的执行资源进行定量的一种很自然的方法是为其入侵检测引擎的 CPU 占用率设置上限(超过即 FEDUP)和下限(不足即 HUNGRY)以表明其运

行时的负载。这两个参数值的设置取决于运行 PeerIDS 实例的计算机的性能, 可以通过在启动时运行一段特定的性能测试程序并根据其运行结果由 PeerIDS 系统自动设置。

IP Head	UDP Head	Source pid	A piece of Snort rule
---------	----------	------------	-----------------------

图 3 一种用于 Snort 包装器的 Workload 消息

6 结 论

通过把网络入侵检测工作的各正交子集分布到点对点等网络中, PeerIDS 系统具有较高的可靠性。同时由于整合了众多的 PeerIDS 实例, 系统提供了强大的人侵检测性能。PeerIDS 系统的可扩展性主要表现在: 只要遵守与系统中运行的人侵检测 Workload 相一致的接口, 对新种类网络攻击进行检测的工作将很容易加入到系统中来。

为提高系统的通讯效率, 未来拟在 PeerIDS 的通讯中采用认证机制来强化其安全性并将通过实现名声 (Reputation) 机制和在 Peer List 中根据各 Peer 的 Reputation 和 Activeness 应用优先级列表, 具体仍待后续研究。

参考文献:

- [1] 董晓梅, 王丽娜, 于戈. 分布式入侵检测系统综述[J]. 计算机科学, 2002, 29(3): 16-19.
- [2] 彭志豪, 李冠羽. 分布式入侵检测系统研究综述[J]. 微电子学与计算机, 2006, 23(9): 191-196.
- [3] 蔡晨, 王泽兵, 冯雁, 等. 基于 Super-Peer 的对等网络研究[J]. 计算机应用研究, 2004, 21(6): 258-260.
- [4] 王伟平, 罗熹, 王建新. SACS: 一种可扩展的匿名通信系统[J]. 小型微型计算机系统, 2007, 28(2): 237-242.

(上接第 172 页)

文献[7]算法比较, 鲁棒性有了明显的提高。

文中提出的信息隐藏算法不能抵抗同步攻击和 D/A, A/D 转换操作, 这是下一步算法研究改进的重点。

参考文献:

- [1] Cooperman M, Moskowitz S. Steganographic Method and Device[P]. USA: [s.n.], 1997.
- [2] Kim H J, Choi Y H. A novel echo hiding scheme with backward and forward kernels[J]. IEEE Trans. Circuits Syst. Video Technol., 2003, 13: 885-889.
- [3] Lie Wen-Nung, Chang Li-Chun. Robust and high-quality time-domain audio watermarking based on low-frequency

amplitude modification[J]. IEEE Transactions on Multimedia, 2006, 8(1): 46-59.

- [4] Seok Jong won, Hong Jin woo, Kim Jin woong. A novel audio watermarking algorithm for copyright protection of digital audio[J]. ETRI Journal, 2002, 24(3): 181-189.
- [5] 马翼平, 韩纪庆. DCT 域音频水印: 嵌入对策和算法[J]. 电子学报, 2006, 34(7): 1260-1264.
- [6] 何琴, 邹华兴, 白剑. 基于小波变换的语音信息隐藏算法[J]. 计算机应用研究, 2005(12): 118-119.
- [7] 王向红, 赵红, 崔永瑞. 一种新的混合域自适应数字音频水印算法[J]. 小型微型计算机系统, 2006, 27(2): 316-319.
- [8] 韩纪庆, 张磊, 郑铁然. 语音信号处理[M]. 北京: 清华大学出版社, 2004: 27-30.