基于轻负载代理的协同分布式入侵检测系统*)

张 琨 刘凤玉
(南京理工大学计算机科学与技术系 南京210094)

Lightweight Agent for Collaborative Distribution Intrusion Detection System

ZHANG Kun LIU Feng-Yu
(Department of Computer Science and Technology, NJUST, 210094, Nanjing, PRC)

Abstract The LAFCDIDS (Lightweight Agent for Collaborative Distribution Intrusion Detection System) presented in this paper is a distributed intrusion detection system with the ability of collaborative detection in real time. The hierarchy architecture of agents and the ability of collaborative detection in real time are evident characteristics of the LAFCDIDS. Lightweight agent and agent sensitivity are LAFCDIDS's new concepts, which can reduce the overload of protected system, shorten the period of intrusion detection, and are suitable for monitoring the distributed collaborating attacks.

Keywords IDS(intrusion detection system), Agent, Dynamic aggregation, Distributed intrusion detection, Information security

1 引言

网络安全在Internet中的重要性越来越明显,因此,作为网络安全主动防御措施的入侵检测系统(IDS,Intrusion Detection System)也越来越受到关注[2,6]。当前,智能化、分布式的入侵检测系统以其检测效率高、容错性强和易配置性受到研究人员的青睐[2]。但是现有的分布式入侵检测系统还存在以下缺点[1,3]:

1. 大多数分布式入侵检测系统只是在数据采集上实现了分布式,数据的分析、入侵的发现还是由单个程序完成的,系统可扩展性差;

2. 由于检测部件依赖于数据收集部件,如果数据收集部件或检测部件出错,都会影响系统的正常运行,系统存在单点失效问题;

3. 目前许多基于代理的分布式入侵检测系统,代理间的通信都是纵向的,即信息从负责收集数据的代理传送到上层代理或监控器,同层代理间不具备通信协作能力,只有当信息到达上层代理后才执行分析检测,因此有碍于分布式入侵的提早发现,系统实时检测能力难以满足需求;

4. 由于加入代理而使入侵检测系统的开销增大,干扰了受保护系统的正常工作,降低了系统效率。

针对目前分布式入侵检测技术的缺点,本文提出了一种基于轻负载代理的协同分布式入侵检测系统LAFCDIDS。该系统将入侵检测中的代理层次化,同时利用动态聚合技术实现了轻负载代理和轻负载代理间的通信机制。轻负载代理和代理敏感度(Sensitivity)是本系统提出的新概念。轻负载代理是指自身功能可动态增加或删除的代理,它的提出主要是为了减少检测系统开销对受保护系统的影响。同时,为了增强系统检测分布式入侵的能力和缩短检测时间,本文为轻负载代理增加了敏感度特性。具体实现详见2.3节。

2 基于轻负载代理的协同分布式入侵检测系统

2.1 整体结构

如图1所示,LAFCDIDS采取无控制中心的分层代理结构,每台主机上可以有多个具有不同功能的代理,代理之间相互通信、协作。

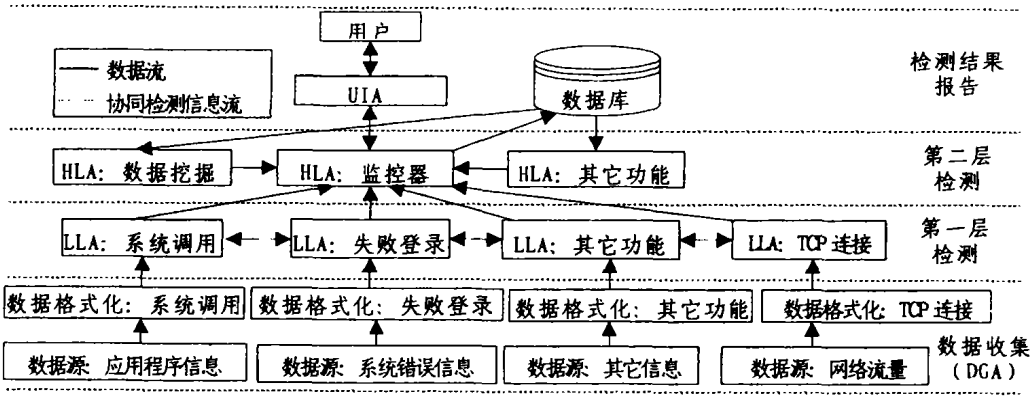

图1 协同分布式入侵检测系统结构

*)受国家自然科学基金(编号:69973020)和国防科工委应用基础基金(编号:J1300D004)资助。张 琨 博士研究生,研究方向为入侵检测系统与信息安全。刘凤玉 博士导师,研究方向网络信息安全。

LAFCDIDS 按功能划分为四个部分：数据收集、第一层检测、第二层检测和检测结果报告。具体包括四类代理：数据收集代理 DGA(Data Gathering Agent)、低层检测代理 LLA(Low Level Agent)、高层检测代理 HLA(High Level Agent)和用户界面代理 UIA(User Interface Agent)。

Java 语言具有平台独立性、安全性和开发速度快等特点，因此本系统选用 Java 作为开发语言。ObjectSpace 公司开发的 Voyager Object Request Broker(ORB)[4]具有 Java 语言本地化和支持移动代理特性，因此选择它实现本系统的部分代理，如 LLA。LAFCDIDS 原型采用 JDK1.2[5]和 Voyager ORB 第3版实现。

2.2 DGA

最底层的代理位于单个主机上，这类代理收集并格式化主机的某类活动信息，我们把这类代理叫做数据收集代理 DGA。每台主机可以拥有多个 DGA 分别监视不同的活动。DGA 从各种形式的数据源收集信息，如系统日志、审计数据、操作统计和网络数据包，并将这些原始信息转化成统一的格式提交给其上层代理 LLA 以作入侵检测分析。

例如本系统中的 DGA：DGFailedLoginAgent，它读取系统日志信息，查找并分析与登录失败相关的记录。当失败登录发生时，由于它记录了用户登录时使用的账号和失败登录事件发生的主机，因此可用它描述登录对象(Login Object)。

2.3 LLA

每台主机上还有一个叫做低层代理 LLA 的移动代理，它是入侵检测系统的第一道防线。LLA 每隔一段时间就转移到与其相关的 DGA 处获取近期从系统中收集的信息，并对这些信息进行分类以检测系统是否受到攻击，最后再将处理后的信息发送到 HLA 处。在 LLA 层，能够发现涉及到多台主机、与网络有关的入侵活动，提高了系统的检测效率，防止入侵对系统的深入破坏。

如本系统中的移动代理：FailedLoginAgent，它访问每台主机的 DGFailedLoginAgent 以获取近期的失败登录记录。在整个分布式系统中，如果在特定时间间隔内收集到的失败登录次数超过阈值，那么失败登录将被标记为攻击者对系统的试图攻击。

2.3.1 轻负载代理和代理敏感度

本系统的特色就在于 LLA。针对第一节提及的当前分布式入侵检测系统的局限性，本系统采用动态聚合技术实现了轻负载代理(即 LLA)及 LLA 间协同检测入侵的能力。

轻负载代理和代理敏感度是本系统提出的新概念。轻负载代理即指代理可按需动态加载功能，打破传统的在代理创建时令其具备所有功能，而不管这些功能在检测中是否会用到。例如随后介绍的敏感性功能，只有执行协同式入侵检测的 LLA 才需具备该功能，而其它代理根本不需要，因此，在检测系统运行过程中，LLA 可根据需要动态加载敏感度功能。轻负载代理的优点是：代码量更少；实现更简单；移动代理的传输速度更快；系统功能的扩展可通过代理的更新实现，与目前的增加新功能代理方法相比，更灵活，实现更简单。

代理的敏感度是为了解决分布式协作入侵而引入的，采用动态聚合技术实现。代理的敏感度影响其对事件的分类(即是否为入侵)。例如，就登录失败事件来说，单机上的少数登录失败事件的原因难以确定，它可能是正常事件(如用户忘记密码或敲错键盘)，也可能是入侵者对系统的攻击(如攻击者利用少量典型的密码尝试登录主机)。然而，如果在检测到登录失败事件前，系统已经检测到端口扫描事件(入侵者往往根据端口扫描确定攻击的目标主机)，那么，LLA 的敏感度功能就能将第一个事件(端口扫描)和第二个事件(登录失败)结合考虑检测入侵。弥补了以往分布式系统难以检测分布式入侵的局限性。

2.3.2 代理敏感度的实现

Voyager ORB 支持动态聚合技术。动态聚合技术提供实时为原始对象附加 facets 的功能，聚合后的原始对象和 facet 作为一个整体存在、移动或删除。图2所示为 LLA(原始对象)与敏感度 facet 的动态聚合。

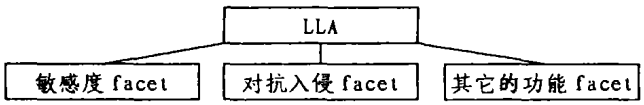

图2 LLA 与敏感度 facet 的动态聚合

如图2所示，随着研究的进一步发展，还可为代理动态增加新的 facet，如入侵对抗 facet，以实时确定入侵点，实现系统功能的动态扩展，提高系统的检测能力。为实现代理敏感度，在 facet 中添加了两个对象(Object)：接收信息 Object 和报告信息 Object。接收信息 Object 的功能是：监听其它 LLA 报告信息 Object 发送的入侵消息，并根据这些信息调整自身 LLA 的敏感度。报告信息 Object 的功能是：当所处的 LLA 判定某事件为入侵时，向其它感兴趣的 LLA 的接收信息 Object 发送入侵信息。图3所示为代理间协同检测入侵的原理。

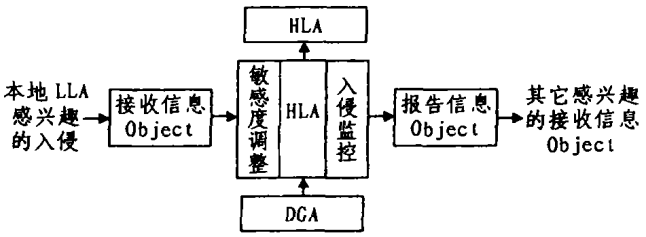

图3 实现 LLA 敏感度特性的 facet

LLA 中报告信息 Object 的信息发送目标(即其它 LLA 的接收信息 Object)不是任意选定的，即只有与某入侵事件相关的代理间才有必要通信。首先应确定每个执行协同入侵检测的 LLA 与其感兴趣的入侵间的相关性，其感兴趣的入侵由其它 LLA 负责监控。下面仅举例说明其间的关联性，如表1所示。

表1 LLA 代理与其感兴趣的入侵

LLA	感兴趣的入侵(其它 LLA 监控)	入侵原理
不被信任的连接	攻击网络端口监督程序改变的配置文件	在攻击完网络端口监督程序或改变配置文件后，攻击者往往会尝试通过网络登录主机
关键文件	攻击网络端口监督程序；NFS	攻击者往往更改某些关键文件以便执行更深入的攻击
失败的登录	端口扫描	攻击者扫描可能的网络端口以发现可以登录的主机

原型系统中每个 LLA 中都包含下述方法：

register()	定义本地 LLA 可检测的事件；
setIntrusionClasses()	定义感兴趣的其它 LLA 上发生的入侵；
getSensitivity()	获得当前 LLA 的敏感度；
sendIntrusionMessage()	向其它接收信息 Object 发布与入侵相关的消息；
recvIntrusionmessage()	从其它报告信息 Object 接收入侵消息，并根据事件的可疑度调整 LLA 的敏感度。

图4所示为利用敏感度 facet 实现的协同分布式入侵检测系统的一部分。NetTCPAgentSensitivity 是基本 Sensitivity 的扩展，它负责监听入侵活动，其中包括文件改动和缓冲区溢出攻击。如果其它代理已经发现有上述入侵活动，那么如果随后 NetTCPAgent 附属的主机也收到异常 TCP 连接请求，该请求有可能是入侵过程中的一部分，那么 NetTCPAgent 将提高自身的敏感度。根据 Voyager 提供的动态聚合技术和 facets 添加规则，为 NetTCPAgent 增加敏感度 facet：NetTCPAgentSensitivity facet。

FailedLoginAgentSensitivity facet 负责监控从 netTCPAgent 发送过来的入侵活动。如果 NetTCPAgent 检测到诸如 telnet 端口扫描的活动时，FailedLoginAgentSensitivity facet 将提升自身的敏感度。因为入侵者往往先扫描可访问的 telenet 端口，然后连接到其发现的 telent 端口，并尝试用典型的口令和熟知的账户登录主机。如果上述攻击发生，那么 FailedLoginAgentSensitivity facet 通过降低对失败登录次数的接收阈值即可检测攻击。

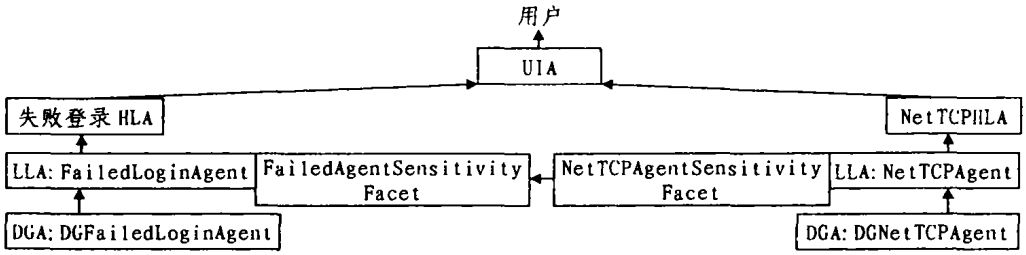

图4 入侵检测的部分示意图

2.4 HLA

高层代理 HLA 着眼于整个网络，它进一步综合分析来自各主机的信息，并将这些信息存入数据库，最后得出最终结论，并提交给 UIA。除了收集和分析 LLA 报告的数据外，HLA 还可创建和删除 LLA，实现对整个系统的管理。为了减少单点失效的风险，HLA 可以有多个，分别位于不同的主机上。在保证数据一致的前提下，一个 LLA 可向多个 HLA 提交报告。这样当一个 HLA 失效时，其它 HLA 仍能正常工作。在图1中，还引入了数据挖掘 HLA，主要是为了在进一步的研究中对数据库中的数据应用数据挖掘算法，以便将从各代理处获得的事件联合考虑，完成对分布式协同攻击的二次检测，提高系统检测的正确性。

2.5 UIA

用户界面代理 UIA 负责管理入侵监控系统和向用户报告入侵。其中包括入侵检测系统用户界面的实现。任何入侵检测系统如果不具有人机交互友好性，那么它将毫无用处。通常用户界面的设计都面临一个共同的问题：如何将高速、分布、持续运作的计算机系统与用户相连。

传统的用户界面是一个显示主机列表的窗口，显示的主机全部是监控主机，用户可以了解任意主机的详细情况，但是系统提供的详细信息非常有限，仅给出了分布式 IDS 组件的基本形式控制。本系统采用多种不同的用户界面，如图形用户界面 GUI(Graphical User Interface)用于入侵检测系统的交互式访问，而命令行界面用于维护报告函数操作。

3 通信机制

代理之间的通信是一个非常重要的问题，如果通信受到了损害，入侵检测系统就会瘫痪。为了保证入侵检测系统和被检测主机、网络的正常运转，代理通信必须做到：不能过分加重主机和网络的通信负担，从而不影响主机和网络的正常运作。

为了使消息能够被多种异构的代理正确理解，本系统对消息进行了统一的格式编排，采用基于 ASN.1语法的描述串。ASN.1入侵层次的实例如下所示：

```
host    在主机上识别入侵
    host.priv-prog   对特权程序的入侵
    host.auth   与认证相关的入侵
net    与网络服务相关的入侵
    net.ip   通过 IP 网络协议发起的入侵
    net.ip.icmp   通过 ICMP 协议发起的入侵
    net.ip.udp   通过 UDP 协议发起的入侵
        net.ip.udp.service=nfs   通过 NFS 发起的入侵
    net.ip.tcp   通过 TCP 协议发起的入侵
        net.ip.tcp.service=login   通过 rlogin 协议发起的入侵
```

系统中的代理通过特定的前缀在系统中标识其感兴趣的入侵。例如，监控所有与 TCP 相关入侵的代理需标识 net.ip.tcp。

结束语 目前的分布式入侵检测系统具有单点失效、可扩展性和实时检测能力差等缺点，而 LAFCDIDS 利用动态聚合技术则能较好地解决上述问题。在本文提出的系统中，分布于整个网络的各类代理相互协作，共同完成入侵检测任务。轻负载代理和代理敏感度是本系统的关键技术。它们为系统提供了实时协同检测分布式入侵的能力，这正是本系统的重要特点。在随后的测试中，LAFCDIDS 的可扩展性和实时检测能力相对于其它系统有了较大的提高，能够检测到如下入侵：在多台机器上尝试多个口令的失败登录，异常的 TCP 网络连接，关键文件的改变，对 sendmail 邮件传输代理的攻击。进一步的工作重点是在 HLA 层为系统引入新的技术，如数据挖掘和数据融合；在 LLA 层增加新的 facet 以实现对抗入侵等功能，进一步提高入侵检测系统的综合性能。

参 考 文 献

1 Moounji A. Rule-Based Distribution Intrusion Detection [D]. Belgium Grandagnage : Institute d'Informatique University of Namurrue,1977

(下转第73页)

钥，但事实上 I 也有 $N_A \oplus N_I$。

2.3 一个改进的 AKA 协议

只要在消息5中 B 的签名域上加上他的长期和短期公钥的 HASH 值就可以挫败此攻击。

消息1 $A \rightarrow B: K_A$

消息2 $B \rightarrow A: K_{Bh}, K_{Bs}$

消息3 $A \rightarrow B: \{N_A, A\}_{K_{Bs}}$

消息4 $B \rightarrow A: \{N_B\}_{K_A}$

消息5 $B \rightarrow A: \{\{N_A, H(K_{Bh}, K_{Bs})\}_{K_{Bh}^{-1}}\}_{K_A}$

消息6 $A \rightarrow B: \{H(N_B)\}_{K_{Bs}}$

这样由 $H(K_{Bh}, K_{Is}) \neq H(K_{Bh}, K_{Bs})$ A 就可以发现错误，使得攻击者 I 无法欺骗 A。

3. 安全登录连接(SSH)协议

3.1 协议描述

SSH 协议是草拟的 Internet 安全协议中的一个认证与密钥分配协议，它也有几个变体[5]，这里我们仅考察服务器 B 利用公钥认证客户 A 的情况。这种情况下协议是：

消息1 $A \rightarrow B: N_A$

消息2 $B \rightarrow A: N_B$

消息3 $B \rightarrow A: K_{Bh}, K_{Bs}$

消息4 $A \rightarrow B: \{\{H(previous.msge), k\}_{K_{Bs}}\}_{K_{Bh}}$

消息5 $A \rightarrow B: \{A, K_A, \{H(A, N_A, N_B)\}_{K_A^{-1}}\}_{K'}$

传递这些消息是为其后 A 和 B 之间通信建立会话密钥 K。在消息1和2中，A 和 B 交换随机数 N_A 和 N_B；消息3中 B 提供他的长期公钥和短期服务公钥 K_{Bh} 和 K_{Bs}；消息4中，A 把会话密钥 K 用 B 的两个公钥加密后传给 B，其中包含先前传递消息的 HASH 值；消息5中，A 提供他的身份 A 及公钥 K_A，并对他的身份及随机数 N_A 和 N_B 的单向 HASH 值用他的私钥 K_A^{-1} 签名，这些消息都用他们其后要共享的密钥 K` 加密，且 K` 可从 K、N_A 及 N_B 导出。通过这个协议的运行服务器 B 对用户 A 的身份获得认证，同时建立了一会话密钥 K`。

3.2 协议上的一个攻击

我们同样按照上述7种运行模式逐一执行协议，在其中模式(2)A⇒I;I⇒B 上执行 SSH 协议发现了一个攻击者 I 既作协议引发者又作响应者的两次运行攻击：

消息1 $A \rightarrow I: N_A$

消息1' $I \rightarrow B: N_A$

消息2' $B \rightarrow I: N_B$

消息2 $I \rightarrow A: N_B$

消息3' $B \rightarrow I: K_{Bh}, K_{Bs}$

消息3 $I \rightarrow A: K_{Ih}, K_{Is}$

消息4 $A \rightarrow I: \{\{H(previous.msgs), K\}_{K_{Is}}\}_{KIh}$

消息4' $I \rightarrow B: \{\{H(previous.msgs), K\}_{K_{Bs}}\}_{KBh}$

消息5 $A \rightarrow I: \{A, K_A, \{H(A, N_A, N_B)\}_{K_A^{-1}}\}_K$

消息5' $I \rightarrow B: \{A, K_A, \{H(A, N_A, N_B)\}_{K_A^{-1}}\}_{K'}$

在该攻击中，入侵者 I 是网络中一合法服务器且具有长期和短期公钥 K_{Ih}, K_{Is}，它能模仿 A 与 B 会话，而且只要在 A 和 I 开始一个正常会话时该攻击就是可能的。当 A 和 I 开始会话时，I 立刻开始和 B 会话以确保两个会话中随机数是一样的，从而 I 能在和 B 的会话中使用 A 的签名消息。这样 B 认为他和 A 使用会话密钥 K 开始了一个会话，事实上 I 也有此会话密钥 K，而且 A 也不会立刻发现错误。

3.3 一个改进的 SSH 协议

为了抵抗此攻击，我们可以在 A 的签名域中增加一些元素，如 B 的身份、证书、公钥或会话钥 K 的 HASH 值。

消息1 $A \rightarrow B: N_A$

消息2 $B \rightarrow A: N_B$

消息3 $B \rightarrow A: K_{Bh}, K_{Bs}$

消息4 $A \rightarrow B: \{\{H(previous.msgs), K\}_{K_{Bs}}\}_{K_{Bh}}$

消息5 $A \rightarrow B: \{A, K_A, \{H(A, B, H'(K), N_A, N_B)\}_{K_A^{-1}}\}_{K'}$

这样 I 就不能在和 B 的会话中使用 A 的签名消息，从而不能欺骗 B。

结束语 我们运用双方密码协议的运行模式分析了两个认证与密钥分配协议，从中可以发现这两协议有一共同弱点，就是签名域没有包含足够的信息，从而留下安全隐患。实际上，这也是一类利用签名进行认证的协议设计共有的问题，仔细设计签名域以抵抗攻击者的假冒应该是这类协议设计遵循的一个原则。

参 考 文 献

1 Lowe G, Rosse B. Using CSP to detect errors in the TMN protocols. IEEE Transactions on software engineering. 1997, 23(10): 659～669

2 Lowe G. Breaking and fixing the needham-schroeder public-key protocol using FDR. In: Proc. of TACAS(Tools and Algorithms for the Construction and Analysis of Systems), vol. 1055, Springer Verlag, 1996. 147～166

3 Zhang yuqing, Li Jihong, Xiao Guozhen. An approach to the formal verification of the two-party cryptographic protocols. ACM Operating System Review, 1999, 33(4): 48～5

4 Safford D, Schales D, Hess D. Texas A&M University anarchistic key authorization. In: Proc. of the Sixth Usenix Security Symposium, New york, 1996. 179～185

5 Ylonen T. SSH--secure login connections over the internet. In: Proc. of the Sixth Usenix Security Symposium, New york, 1996. 37～42

(上接第68页)

2 Honavar V, Miller L, Wong J S K. Distributed knowledge networks. In: Proc. IEEE Information Technology Conf. Syracuse, NY, USA, Sep. 1998. 87～90

3 Jansen W, Mell P, Karygiannis T, Marks D. Mobile agents in intrusion detection and response. In: Proc. of the 12th Annual Canadian Information Technology Security Symposium, Ottawa, Canada, June 2000

4 ObjectSpace, Inc., Dallas, TX. ObjectSpace Voyager Core Technology User Guide, Version 3.0.0, 1999

5 Sun Microsystems. Java Development Kit Version 1.2.x. Online, September 2001. http://www.javasoft.com/products/jdk/1.2.

6 蒋建春，马恒太，任党恩，卿斯汉. 网络安全入侵检测：研究综述. 软件学报，2000, 11(11): 1460～1466