ISSN 1000-9825 Journal of Software 软件学报 2000,11(10):1312~1319

基于 Agent 的分布式入侵检测系统模型*

马恒太 蒋建春 陈伟锋 卿斯汉

(中国科学院软件研究所信息安全国家重点实验室 北京 100080)
(中国科学院信息安全技术工程研究中心 北京 100080)
E-mail: mht@ercist.iscas.ac.cn

摘要 提出了一个基于 Agent 的分布式入侵检测系统模型框架.该模型提供了基于网络和基于主机入侵检测部件的接口,为不同 Agent 的相互协作提供了条件.在分布式环境中,按照系统和网络的异常使用模式的不同特征和环境差异,可利用不同的 Agent 进行检测,各 Agent 相互协作,检测异常行为.该模型是一个开放的系统模型,具有很好的可扩充性,易于加入新的协作主机和入侵检测 Agent,也易于扩充新的入侵检测模式.它采用没有中心控制模块的并行 Agent 检测模式,各 Agent 之间的协作是通过它们之间的通信来完成的,各 Agent 之间可以交流可疑信息和进行数据收集.Agent 之间各自独立,相互协作,合作完成检测任务.另外,模型采用一定的状态检查和验证策略,保证了 Agent 的自身安全和通信安全.该模型与特定的系统应用环境无关,因此,提供了一个通用的入侵检测系统框架模型.

关键词 入侵检测,分布式模型,Agent,通信,协作.
中图法分类号 TP393

随着网络革命的到来,Internet 在为发展带来巨大机会与可能性的同时,也带来了恶意入侵的风险.为了保护系统资源,需要建立安全检测机制以发现未授权入侵和破坏.这个研究领域就是入侵检测(intrusion detection,简称 ID).为此目的而设计的系统称为入侵检测系统(IDS).

计算机系统应能提供保密性、完整性以及抵抗拒绝服务攻击的能力.攻击者一般是利用操作系统或者应用程序的缺陷来攻击系统的.

目前,对付破坏系统企图的一个实用方法是按照一定的安全策略为系统建立相应的安全辅助系统,IDS 就是这样的一类系统.如果系统遭到攻击,IDS 可以尽可能地检测到,甚至是实时地检测到,然后采取恰当的处理措施.IDS 所起到的作用是安全触发器的作用,通过检测入侵事件,就可以及时阻止事件的发生和事态的扩大.

网络的迅速发展使得现在的网络资源一般都是分散分布的.一般一个单位就是一个局域网,通过网关与 Internet 相连,在内部共享资源.网关成为整个局域网的安全集中点,内部机器处于同一安全层次,只要有人突破网关,攻破一台机器,就相当于攻破了整个局域网.所以,现在的检测模型也应该是分布式的,由不同的检测实体监控不同的主机和网络部分,大家相互协作完成检测任务.特别是信任主机之间更要相互协作,因为攻击者的下一个目标最有可能就是信任主机.本文的符号系统沿用文献[1].

1 传统的入侵检测模型

传统入侵检测模型大体上可以分为基于误用的入侵检测模型和基于异常的入侵检测模型两种.

* 本文研究得到国家重点基础研究发展规划项目(No.G199035810)和中国科学院软件研究所青年创新基金(No.cx2k5606)资助.作者马恒太,1970 年生,博士生,主要研究领域为网络信息安全,分布式计算.蒋建春,1971 年生,博士生,工程师,主要研究领域为计算机网络,信息安全.陈伟锋,1976 年生,硕士生,主要研究领域为入侵检测,智能卡技术.卿斯汉,1939 年生,研究员,博士生导师,主要研究领域为信息安全理论与技术.

本文通信联系人:马恒太,北京 100080,中国科学院软件研究所信息安全中心

本文 2000-05-31 收到原稿,2000-06-30 收到修改稿

1.1 误用入侵检测模型(misuse detection model)

误用入侵检测指的是通过预先精确定义的入侵模式对观察到的用户行为和资源使用情况进行检测.入侵签名(signature)说明了导致误用事件的弱点的特征、条件、序列和关系,还包含系统状态.入侵签名对检测入侵是非常有用的.误用入侵检测器根据输入事件的模式来检测入侵.

1.2 异常入侵检测模型(anomaly detection model)

异常入侵检测是从审计记录中抽取出一些相关量(measure)进行统计,为每个用户建立一个用户扼要描述文件(profile).轮廓(profile)是用户正常行为的统计概要,当用户行为与他以前所建立的轮廓差异大到一定程度时,就认为可能有入侵行为发生.用户的轮廓可根据具体需要定期修改.异常入侵检测模型使用的方法主要有统计分析和专家系统等.

1.3 传统入侵检测模型的局限性

异常入侵检测的主要前提是入侵活动集作为异常活动的子集.理想的情形是,异常活动集同入侵活动集是一致的.这样,检测异常活动恰恰是检测了入侵性活动.结果不会造成入侵的虚报和漏报.可是,入侵活动集并不总是与异常活动集相符合.这里有两种可能性,每种情况发生的概率都不为零.

- 入侵而不显异常:这属于误否定型错误.
- 异常而非入侵:这属于误肯定型错误.

定义异常的阈值设置偏高,就会导致误否定错误.误否定错误的后果是严重的,它不仅仅是检测不到入侵,而且还会给安全管理员以安全的错觉,这是 IDS 的副作用.但定义异常的阈值设置偏低,就会导致难以忍受的误肯定判断,误肯定太多就降低了入侵检测方法的效率,而且会增添安全管理员的负担,这是因为安全管理员必须调查每个被肯定的事件.

由于 profile 需要常常维护和更新,因此,异常检测器容易导致计算开销增大.

误用入侵检测的主要假设是具有能够被精确地按某种方式编码的攻击.在实践中,某种编码可能无法有效地捕获某些独特的入侵.因此,这种方法主要的局限性是仅仅可检测已知的弱点,对检测未知的可能入侵用处不大.

这种方法的另一个缺点是,不得不考虑实际审计处理.但审计过程中用户级调用读写函数常常与审计跟踪信息不一致(这是由于读写缓冲区所引起的),这样就易于造成误否定判断.

这种方法也是以事件数据的完整性为前提的,因而不能可靠地检测到掩盖了某些特征的具有欺骗性的攻击.另外,有的破坏方法可能找不出来源,就像 wiretapping,它们不产生可检测的特征,不能够直接被检测到.

对比这两种入侵检测方法可发现,异常检测难于定量分析.这种检测方式有一种固有的不确定性.与此不同的是,误用检测会遵循定义好的模式,它们能通过对审计记录信息作模式匹配来检测,但仅可检测已知的弱点.

1.4 已有的分布式入侵检测系统

通过对传统入侵检测模型的分析,我们可以看到,各种检测方法模型和技术手段都有其局限性,没有比较通用的检测方法.一个解决方法就是对不同的检测环境进行分类,对不同的环境采用不同的检测方法和技术手段.这就是分布式入侵检测的思想,它采用多个检测部件,各检测部件选用不同的检测方法,协同合作,完成检测任务.这有利于取各种检测方法之长,以大幅度地提高检测效率和准确性.

到目前为止,已有一些研究机构对分布式入侵检测进行了有益的研究,也建立了一些实验性系统.UC Davis 的 GrIDS[7]在每台主机上运行一个模块,它们向一台固定的机器发送信息,然后在这台机器上建立网络活动的图形描述,用这个描述来检测可能的入侵.NADIR[3]系统使用已有的服务节点在 Los Alamos 国家实验室的集成计算机网络(ICN)上进行分布式审计数据收集,然后由一个中心专家系统来分析这些数据.CSM[4]用于执行分布式入侵检测,它不需要层次组织和协调中心.每一个 CSM 就在它所在的主机上进行入侵检测,但可以和其他的 CSM 交换信息.这种结构也允许 CSM 们在检测到入侵时采取响应.文献[5]提出了模拟生物的免疫系统使用分布式部件执行入侵检测的思想.文献[6]描述了一个能以层次方式组织的分布式部件,但具体的实现细节不清楚.EMERALD[2]提出了入侵检测的分布式结构,在主机上配置服务监视部件进行监视.为了减少数据量,他

们用层次方式定义了几层监视部件.可以对监视部件编程执行一些功能.美国普渡大学设计的AAFID系统原型也采用了分布式部件进行数据收集,各部件是以层次型的管理结构组织的.

由以上介绍可知,现有的分布式检测系统大多是利用分布式部件进行数据收集,收集到的数据最后被传送到一个处理中心,在处理中心进行统一处理.这种结构使检测系统无法达到实时性.而且,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.为了解决上述问题,我们提出了基于Agent的分布式入侵检测模型,该模型将入侵检测和实时响应分布化,真正实现了分布式检测的思想.

2 基于Agent的分布式入侵检测模型

2.1 整体结构

我们所提出的入侵检测模型采取无控制中心的多Agent结构,每个检测部件都是独立的检测单元,尽量降低各检测部件间的相关性,不仅实现了数据收集的分布化,而且将入侵检测和实时响应分布化,真正实现了分布式检测的思想.

我们所提出的入侵检测模型是以自治Agent为组织单元,其中有3类自治Agent:入侵检测Agent(IDA)、通信服务Agent(TSA)和状态检查Agent(SDA),如图1所示.

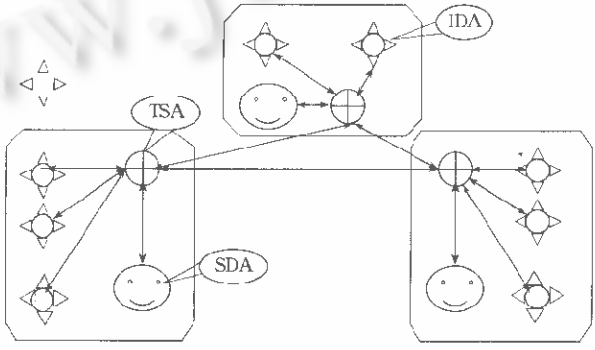


Fig.1 Framework of the model
图1 系统模型框架示意图

2.2 TSA

TSA是专门用于通信服务的Agent,它在每台主机上都是唯一的,是主机内和协作主机间IDA通信的桥梁.当IDA要传送数据时,它指定目标IDA,然后将数据传送给TSA.TSA将数据包转送给目标主机上的TSA,目标主机上的TSA再将数据转送给目的IDA.

TSA是数据传送的中介,它记录了与本机所有的IDA和协作主机TSA的联络方式,可以为数据包提供路由服务.它的任务主要是数据的接收和转发,不具有检测能力和控制能力.它应该可以识别两种包:广播包和定向数据包.广播包是发向除广播源以外的所有IDA的,而定向包则是转发给指定的IDA.如果定向包是发给本地IDA的,TSA就直接转发给它;如果是转发给协作主机上的IDA的,就转发给协作主机的TSA,由协作主机上的TSA转发给相应的IDA.

2.3 SDA

SDA是Agent进行自身保护和验证的专门Agent,它在每台主机上也是唯一的.它定时检查协作主机的TSA和本机内IDA的状态,并负责向系统管理员报告.

SDA是为保证TSA和IDA的安全而设的.由于每一个目标主机只有一个TSA,它就成为整个安全系统的安全重点,一旦TSA被破坏,整个目标主机中的IDA之间和主机之间就无法进行协作.因此,检查和保证TSA的正常运行状态是极其重要的.SDA能定时检查协作主机TSA的状态,一旦发现某台机器的TSA活动异常,就向其他机器的SDA查询该机器的状态,如果其他机器的SDA认为该TSA正常,就进行再次查询;如果其他机器的SDA也报告说不正常,就通知系统管理人员,请求检查问题的所在.

另外,它也定时检查本主机上 IDA 的状态.如果发现状态有问题,就通知本机的 TSA,暂停与该 IDA 间的通信,并通知系统管理人员.

它还提供用户接口,用户可以通过它观察或查探各 IDA 的状态,也可以通过它动态地配置 IDA.一个管理人员可以通过检查 SDA 所保存的状态信息来了解系统的运行情况,由于 SDA 所保存的信息是定时修改的,所以可以很容易地了解系统内部各 IDA 和协作主机 TSA 的状态变化情况,为管理人员能进行有效的管理提供了极大的方便.

2.4 IDA

IDA 是本模型的基本检测单元,它们分布在主机和网络各处,每个 IDA 独立承担一定的检测任务,检测系统或网络安全的一个方面.在模型中,各 IDA 有独立的数据源、运行模式和响应方式,各 IDA 之间进行相互协作,对系统和网络用户的异常或可疑行为进行检测.不同的 IDA 按照检测环境的不同,采用不同的检测方法和技术.

在本系统中,各 IDA 的行为模式是很相似的,一般都经过以下几个步骤(如图 2 所示):

(1) 截获系统或网络信息,作日志;

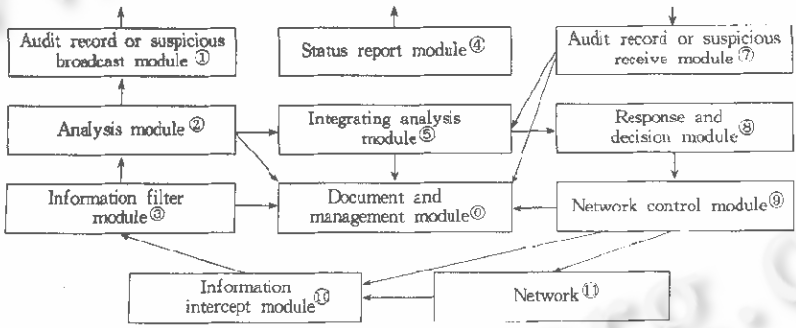
(2) 进行审计分析(模式匹配或异常越界检查,并对相应事件进行响应);

(3) 进行数据处理,确定可疑度;

(4) 与其他 IDA 通信,对可疑级别达到一定程度的事件进行可疑广播;

(5) 如果需要将数据传送给其他 IDA,就通过 TSA 将数据送出;

(6) 记录用户信息.



①审计数据可或疑广播模块,②分析模块,③信息过滤模块,④状态报告模块,⑤综合分析模块,⑥档案记录管理模块,⑦审计数据或可疑接收模块,⑧反应决策模块,⑨网络控制模块,⑩信息截取模块,⑪网络.

Fig. 2 IDA processing flow

图2 IDA处理流程

在图 2 中,状态报告模块是以其他模块的状态报告作为数据来源的,但为简洁起见,在图中没有标出该模块和其他模块的关系.

2.4.1 IDA 检测数据的获取

入侵检测首先需要对系统作日志和审计,而日志和审计涉及到登记和分析系统状态,计算机系统的特点是通过实体集 E 和活动集 A 来表现的.E 是系统的组成部分和各自的状态,A 是引起系统状态变化的事件集合.对所有的 $e\in E$,$VAL(e)$ 是与实体 e 相关的状态集合,$VAL(E)=\{VAL(e)|e\in E\}$.字符串集合 N_E,N_V,N_A 用于命名集合 E,$VAL(e)$ 和 A 中的实体,$h_E:E\rightarrow N_E$,$h_V:VAL(E)\rightarrow N_V$ 和 $h_A:A\rightarrow N_A$ 是 3 个命名函数.为了描述方便,我们指定集合 A 中包含空事件.

定义 1. 系统状态 s 是一个一元组(E).所有可能状态的集合 S 组成状态空间.系统相关状态 $\sigma\subseteq s$ 是 s 中要考虑的部分.Σ 是相关状态空间,$\Sigma=\{\sigma|\sigma\subseteq s\wedge s\in S)\}$.

定义 2. 系统是一个四元组(A,S,s_0,T),其中 S_0 是初始状态,$T:A\times S\rightarrow S$ 是系统迁移.T 是反映状态改变的映射函数.在系统的生命期内,一系列映射函数 T 将被执行,系统状态将随之而变化.

定义 3. 设 N 是自然数集,系统历史函数 $\Pi:N\to A\times S$,其中 $\Pi(0)=(a_0,s_0)$,s_0 是系统的初始状态.

$$\forall n\in N[\Pi(n)=(a,s)\wedge\Pi(a+1)=(a^*,s^*)]\to s^*\in T(a,s).$$

相关状态历史是函数 $\Pi:N\to A_\Sigma\times\Sigma$,因此

$$\forall n\in N[\Pi(n)=(a,s)\to\pi(n)=(a,\sigma)].$$

如果 $\Pi(i)=(a,s)$,$\Pi(i-1)=(a^*,s^*)$,我们把使系统状态从 s 变为 s^* 的转换函数 T 的成员记为 T_i.也就是说,$T_i:S\to S$ 和 $T:a\times S\to S$ 是相同的,其中 a 是 $\Pi(i+1)-(a,s)$ 的第 1 个元素.

相关状态转换映射函数 $\tau:A_\Sigma\times\Sigma\to\Sigma$.

定义 4. 如果条件 $\forall i\{T_i(s_{i-1})=s_i\to\exists\tau_i[\tau_i(\sigma_{i-1})=\sigma_i]\}$ 成立,则我们说相关状态空间 Σ 是蕴含的.

也就是说,如果 σ 的元素包含足够的系统状态信息,使得 σ_{i-1} 在 T_i 的一些转换关系下映射为 σ_i,那么在转换关系 T_i 下,下一状态的相关部分只通过查看 σ_i 就可以决定,这样,只有 $\sigma_i\subseteq S_i$ 的那些部分是相关的.如果假定系统状态的相关部分是蕴含的,我们只需处理相关部分就可以了.

日志函数分析和提炼系统状态的相关部分,并把它们转化为输出.

定义 5. $\lambda_{\text{state}}:\Sigma\to N_E\times N_V$,$\lambda_{\text{state}}$ 是状态日志函数.

定义 6. $\lambda_{\text{change}}:A_\Sigma\times\Sigma\to N_A\times N_E\times N_V$,$\lambda_{\text{change}}$ 状态转换日志函数.

合起来,$\lambda=\lambda_{\text{change}}\cup\lambda_{\text{state}}$ 是日志函数.

$\lambda:A\times\Sigma\to O$,$O=N_E\times N_V\cup N_A\times N_E\times N_V$,$O$ 是日志函数的输出集合.

直观上来说,状态日志函数记录相关部件的系统状态,状态变化日志函数记录特殊的活动,这些活动会使系统状态的相关部分发生变化.日志函数的输出是记录系统状态和变化的数据.在一个日志系统中,两类日志函数需要同时发挥作用.

定义 7. 如果存在唯一的事件 a,实体 e 和实体的状态 $VAL(e)$ 使得 $h_A(a)=n_a$,$h_E(e)=n_e$ 和 $h_V(VAL(e))=n_v$.我们就说 $o=(n_a,n_e,n_v)$(或 $o=(n_e,n_v)$)是可逆的.

直观上说,输出是可逆的则意味着实体的状态可从日志信息中得出.

定义 8. 系统日志就是一系列输出 $o_0,o_1,\ldots$,其中 $\forall i\exists j[\lambda(\pi(j))=o_i]$,如果每个 o_i 都是可逆的,那么日志是唯一的.如果日志是唯一的,并且产生输出 $o_i(i\geqslant 0)$ 的相关状态序列就是系统的相关状态历史,那么日志是完备的.

命题. 假定 $\sigma_0,\sigma_1,\ldots$ 是相关状态历史记录,日志 $L=(o_0,o_1,\ldots,o_m)$ 是完备的,当且仅当下列条件成立:

(1) 每个 o_i 是可逆的;

(2) $o_0=\lambda_{\text{state}}(\sigma_0)$;

(3) 对所有 $i\geqslant 1$,$o_i=\lambda(\pi(i))$.

这表示只有变化日志是不够完备的,相对于系统初始状态,还必须进行状态日志.要正确地追踪系统,状态日志部分必须是蕴含的.转换执行后,引起转换的活动和新的状态必须记录.

2.4.2 审计分析

审计包括日志的求精、结果分析和用户或程序的结论.函数 $r:O\to O$ 求精从 λ 输出的日志信息;函数 $a:O\to\Omega$ 分析求精后的日志,输出审计信息序列 $\omega_i\in\Omega$;函数 $n:\Omega\to\Sigma\times\Omega$ 给出结果的恰当分类,可能引起系统修改相关状态部分.为方便起见,我们将它们合并成一个函数 $\alpha:O\to(\Sigma\times\Omega)$.设 $O_i=\{o_k\mid k\leqslant i\}$.如果 $\alpha(O_i)=(\sigma_i,\omega_i)$,那么审计函数不改变系统的状态,也就是说只提供信息.如果 $\alpha(O_i)=(\sigma_j,\omega_j)$,其中 $j\neq i$,审计函数反馈一个信息给系统,修改系统的状态,这是反馈性的.

2.4.3 IDA 间的协作

在我们的系统模型中,一个关键思想是,同时存在许多 IDA,但每个 IDA 都是独立的检测分析单元,没有控制核心,每个 IDA 监控主机或网络安全的一个方面.有时单个 IDA 所收集的信息可能不能判定可疑行为,需要数个 IDA 一起才能涵盖入侵的所有方面.为了正确检测到可疑行为,这些 IDA 必须相互协作来完成检测任务.

2.4.3.1 数据收集

在分布式环境下,一些 IDA 的检测数据是来自多台主机的审计数据,这时就需要相应主机上有 IDA 来完成

数据收集任务.在我们的模型中,数据收集是通过 IDA 间的通信实现的,当一个 IDA 收集到的数据能满足对方的要求或部分要求时,则会将自己所收集的数据中的相应部分发送给对方.

设有 N 个 IDA 为 IDA B 收集数据,如果 A_j 的审计信息输出 ω_{jk} 是和 B 相关的,B 的日志函数输出为 $o_k=\bigcup_{j=0}^{N}\omega_{jk}$,$\omega_{jk}$ 是 A_j 的审计信息序列.IDA B 的日志信息序列集合:$O_i=\{o_k|k\leqslant i\}$,如果 $\forall\ ka_k(O_i)=(\sigma_{ki},\omega_{ki})$,那么审计函数不改变系统的状态,也就是说只提供信息.如果 $\exists\ ka_k(O_i)=(\sigma_{kj},\omega_{ki})$,其中 $j\neq i$,审计函数反馈一个信息给 A_k,修改 A_k 的相关系统状态,这是反馈性的.

2.4.3.2 可疑广播

在我们的模型下,有一种警告广播包.当一个 IDA 接收到这种广播包时,就表示有 IDA 相信可能有与可疑相关的活动发生,希望提醒相关 IDA 注意.该 IDA 将提升自己的可疑度,当该 IDA 进行后续分析时,也可以进行可疑广播.最终,某个 IDA 的可疑度将超过预先设定的阈值,它就可以向操作员发出警告信息,请管理员检查是否发生了入侵事件.

设有 M 个 IDA 相互协作,如果 A_i 的可疑度广播值是 v_{ik},A_i 的可疑度为 $\beta_{jk-1}=\beta_{jk}+\sum_{i=0}^{M}\lambda_i v_{ik}$,$\beta_{j0}=0$,$\lambda_i$ 是加权因子 $0\leqslant\lambda_i\leqslant 1$.

另外,我们引入了一个定时器函数:$\theta:T\to N$,可以让 IDA 的可疑度随时间而下降,直到降到一个规定的下限为止.如果 IDA 接收到一个可疑广播,将提升自己的可疑度.如果没有其他广播来提升,它就会逐渐恢复到正常操作状态,并继续监控.

2.5 基于 Agent 分布式入侵检测模型的优点

(1) 独立性.IDA 是独立运行的程序实体,可以独立开发.在放入具体运行环境前,可以进行独立测试.

(2) 灵活性.IDA 可以独立地启动和停止,也可以进行动态配置,而不影响其他 IDA 的正常运行.要收集新数据或检测新类型的入侵,可以通过对原 IDA 进行重新配置或增加 IDA 来实现.

(3) 系统可扩充性好.无论是增加检测主机,还是在主机中增加 IDA 都简单、方便.

(4) 错误扩散小.如果某个 IDA 出现问题或受到破坏,那么,仅仅与该 IDA 相关的检测部分失效,SDA 会很快检测到它的状态,进行相应的处理.使危害限制在最小的范围内.

(5) 数据来源不受限制.不同的 IDA 可以选用不同的数据源.由于各 IDA 是独立实现的,那么数据来源就可以采用多种形式:审计数据、检查系统配置、捕获网络包或其他合适的来源.

(6) 兼容性.该模型可以既包含基于主机的 IDA,又包含基于网络的 IDA,超越了传统入侵检测模型的界限.

(7) 与平台和开发语言无关.由于 IDA 是独立的,它们可以分别开发,而且可以基于不同平台使用不同的语言开发,只要遵循统一的通信协议和通信格式就可以了.

(8) 协作性.虽然每个 IDA 检测的只是主机或网络安全的一个方面,甚至可能是简单的命令审查,但由于 IDA 之间可以交换信息,就可以在一些 IDA 中产生很复杂的结果.

3 实践活动

3.1 几个典型 IDA

3.1.1 特殊服务检查 IDA

在网络环境中,服务器可以为用户提供各种服务.一些恶意用户会利用服务进程本身的问题或服务之间的漏洞进行破坏活动.为了防止破坏,保护服务器的安全,需要截获用户请求进行安全检查.几个主要的服务检查 IDA 包括:Mail 服务检查 IDA、Telnet 命令检查 IDA、FTP 服务检查 IDA 和 HTTP 服务检查 IDA 等.这类 IDA 采用误用检测模型,对各种服务中已知的漏洞进行分析,建立模式库,将用户行为和模式库中的异常模式进行匹配,一旦发现异常模式,就认为发生了入侵企图和可疑行为.

以 FTP 服务检查 IDA 为例,IDA 对用户按授权表进行权力检查,一旦用户发出对非授权文件进行操作的

命令,IDA就记录该异常事件,并对用户发出警告.

在该IDA中,N_A是FTP相关操作的名字,N_E是文件目录和用户的名字,N_V是保护状态的新设置,日志函数输出FTP相关操作名、被修改的文件名,修改文件的主体名和新的保护状态模式,$O=N_A\times N_E\times N_V$,这是变化日志.如果所有文件的保护模式要定期记录,N_E就是文件和用户名,N_V是保护状态的设置,日志函数的输出包含被扫描的实体名和相关保护状态模式,$O=N_E\times N_V$,因此这是状态日志.对日志信息进行审计分析,可以及时发现攻击行为,甚至可以进行状态恢复.

3.1.2 系统调用分析IDA

该IDA处于目标主机中,截获用户进程的系统调用,按照预先设定的关键操作对这些数据进行审查,一旦发现用户调用了关键操作,就认为用户是可疑的.通过系统调用来检查入侵行为提高了检测的可靠性.

采用系统调用方法,N_A是系统调用的名字,N_E是文件和用户的名字,N_V是保护状态的新设置;日志函数输出系统调用名、被修改的实体名,修改它的实体名和新的保护状态模式,$O=N_A\times N_E\times N_V$,这是变化日志.如果所有文件的保护模式要定期记录,N_E就是文件和用户名,N_V是保护状态的设置;日志函数的输出包含被扫描的实体名和相关的保护状态模式,$O=N_E\times N_V$,这是状态日志.对日志信息进行审计分析,可以及时发现攻击行为.

3.1.3 网络流量统计分析IDA

该IDA处在服务器、包过滤器或网关上,通过检查包头信息,对控制包的流向和数据包的流量进行统计,按照统计数据建立一些用户扼要描述文件(profile),一旦发现新的统计数据和参考profile的差异度超越一定阈值,就认为发生了可疑事件.这类IDA采用基于统计分析的异常检测模型,以群体的行为建立行为模式来匹配个体行为,过分超越于群体之外的个体将被认为是异常的.比如,一般用户的请求包和应答包是成一定比例的,如果发现一个用户的两种包的比率在一定时间内远远大于一般用户,甚至有时只请求,而不响应,则说明用户可能在进行拒绝服务攻击,此时,IDA应记录该异常现象,并提醒管理员注意.

3.2 通信环境

在多Agent系统中,Agent之间的协作完全是通过"通信"来实现的.我们的实验系统采用UDP和TCP相结合的方式来实现通信.其通信内容为一个四元组:

〈通信内容〉::=〈发送者〉〈接收者〉〈时间〉〈数据〉.

从程序设计的层次来看,IDA之间的通信是由事件激发的.在IDA中,当一些特殊事件发生时,IDA就会向TSA发送信息包.TSA接收到信息包,就会根据信息包的目的地进行转发:当目的地是本地时,TSA就直接将信息包发向目的IDA的接收端口,目的IDA接收到信息后,就进行相应的处理;当目的地是协作主机上的IDA时,TSA就将信息包转发给该协作主机的TSA,再由该TSA转发给相应的目的IDA.

由于IDA之间的信息主要是简短的信息,所以,一般信息包都是以UDP包的形式,源IDA直接将信息封装在UDP包中,送往TSA.

我们还提供TCP协议来实现大量数据的传送.TCP协议所处理的对象是文件,当IDA需要向一个远程IDA传送一个文件时,它向TSA发送一个请求,TSA将和对应的TSA协商建立一个TCP链接来传送文件,最后再由对应的TSA将文件转发给目标IDA.例如,审计数据的定时备份,就是采用TCP协议来实现的.

3.3 安全机制

IDA的引入,可能给系统带来额外的安全问题.入侵者可能通过攻破IDA获得很大的系统权限,或通过获得IDA的状态来获得系统和主机的有关信息.为此,我们采取了一些措施,对网上传送的审计数据进行加密处理.这样一来,即使用户侦听到了审计数据也很难获得有用信息.

另外,SDA也会定时探测IDA的状态,对状态异常的IDA,SDA会迅速作出反应,保存现场,通知TSA暂停该IDA的通信服务,并报告系统管理员.

由于本模型的安全机制是一个重要的问题,我们将另文专门讨论,在此仅作简述.

4 结论和讨论

本文给出了一个基于 Agent 的分布式入侵检测系统模型框架.该模型可以与检测基于网络和基于主机的入侵行为的 IDA 接口.它是一个开放的系统模型,具有很好的可扩充性,易于加入新的协作主机和 IDA.本模型采用没有中心控制模块的并行 Agent 检测模式,各 IDA 之间的协作是通过它们之间的通信来完成的.IDA 之间各自独立,相互协作,合作完成检测任务.另外,模型采用一定的状态检查和验证策略,保证了各 Agent 的自身安全和通信安全.尽管我们的实验原型是在 Unix 与 Linux 操作系统下实现,但该模型与特定的系统应用环境无关,因此,提供了一个通用的入侵检测系统模型框架.

本文的目的主要在于模型框架的构造,我们的主要目的是寻找一种更适合于进行入侵检测的体系结构,在此体系下,各检测部件更能保持独立性和发挥自主性.

由于篇幅所限,这里不能给出更多的实现细节,更详细的实现叙述将在后续论文中给出.

参考文献

1 Bishop M. A model of security monitoring. In: Proceedings of the 5th Annual Computer Security Applications Conference. 1989. 46~52. http://seclab.cs.ucdavis.edu/papers.html

2 Staniford-Chen S, Cheung S, Crawford R *et al*. GrIDS: a graph based intrusion detection system for large networks. In: Proceedings of the 19th National Information Systems Security Conference, Vol 1. National Institute of Standards and Technology, 1996. 361~370

3 Hochberg J, Jackson K, Stallings C *et al*. NADIR: an automated system for detecting network intrusion and misuse. Computers and Security, 1993,12(3):235~248

4 White G B, Fisch E A, Pooch U W. Cooperating security managers: a peer-based intrusion detection system. IEEE Network, 1996,10(1):20~23

5 Forrest S, Hofmeyr S A, Somayaji A. Computer immunology. Communications of the ACM, 1997,40(10):88~96

6 Hunteman W. Automated information system alarm system. In: Proceedings of the 20th National Information Systems Security Conference. National Institute of Standards and Technology, 1997

7 Porras P A, Neumann P G. EMERALD: event monitoring enabling responses to anomalous live disturbances. In: Proceedings of the 20th National Information Systems Security Conference. National Institute of Standards and Technology, 1997

Distributed Model of Intrusion Detection System Based on Agent

MA Heng-tai JIANG Jian-chun CHEN Wei-feng QING Si-han

(*State Key Laboratory of Information Security Institute of Software The Chinese Academy of Sciences Beijing* 100080)
(*Engineering Research Center for Information Security Technology The Chinese Academy of Sciences Beijing* 100080)

Abstract The framework model proposed in this paper is a real time intrusion detection system based on Agent, which provides an interface for intrusion detection components. Such interface can be used to detect intrusion behaviors based on both network and hosts. According to the different system or network usage patterns and environment diversity, a set of various agents will be created which cooperate to detect the anomalous aspects. The proposed model is an open system, which has good scalability. It is easy to add new cooperating hosts and agents and to expand new intrusion patterns. agents work in a concurrent way without any central controlling module. The cooperation among Agents is implemented just by communication. Agents are independent but are capable of communicating with each other when they take their actions. The state-checking and policy of authentication mechanism ensure the security of the agents themselves and the communication among them. This model is independent of specific application environment, thus providing a general-purpose framework for intrusion detection systems.

Key words Intrusion detection, distributed model, Agent, communication, cooperation.