入侵检测系统发展的研究综述*)

张相锋　孙玉芳
(中国科学院软件所　北京 100080)

A Survey on the Development of Intrusion Detection System

ZHANG Xiang-Feng　SUN Yu-Fang
(Institute of Software, Chinese Academy of Sciences, Beijing 100080)
Email: xfzhang@sonata.iscas.ac.cn

Abstract　With the fast development of Internet, more and more computer security affairs appear. Researchers have developed many security mechanisms to improve computer security, including intrusion detection (ID). This paper reviews the history of intrusion detection systems (IDS) and mainstream techniques used in IDS, showing that IDS could improve security only provided that it is devised based on the architecture of the target system. From this, we could see the trend of integration of host-oriented, network-oriented and application-oriented IDSs.

Keywords　Information security, Intrusion detection, Intrusion detection evaluation

1. 引言

计算机的飞速发展推动着信息时代的前进。计算机开始时只是作为单机进行一些科学运算，后来发展成为分布式系统，到现在已经进入网络时代，计算机的计算能力有了巨大的提高，同时所处理的业务也扩展到社会生活的各个方面。然而，随着信息交换能力的提高、网络应用范围的不断扩大、Internet 上各类业务的增长，计算机系统和网络的安全问题也日益突出。病毒、木马、黑客等词汇经常在媒体上出现，政府或企业的网站时常被侵袭，不但造成经济上的损失，还对政府和企业的声誉产生了很坏的影响。保障计算机系统和整个网络通信基础的安全已经成为刻不容缓的研究课题。

当前，加强系统安全的技术主要有认证授权、数据加密、访问控制、安全审计、入侵检测等，其中入侵检测技术是为保证计算机系统的安全而设计与配置的一种能够发现并报告系统中未授权访问或异常现象，从而检测违反计算机安全策略行为的技术。

有两种方法来对付入侵者：保护和检测。保护就是构建绝对安全的系统，以防止入侵者攻破系统。然而，实际情况是：

(1)我们不可能构建完全安全的系统。Miller 在其针对 UNIX 系统上流行的应用程序所作的研究报告[15]中指出，没有缺陷的软件只是一个梦想，设计和实现绝对安全的软件系统是极其困难的；

(2)对于全球已经安装的大量的软件，要把它们都转成“安全”的系统也是极其耗时的。因此一个软件的很多版本在同时运行，其中较低的版本往往更容易被攻破；

(3)加密方法有其自身的缺点。密码可能丢失，也可能被破译。如果密码被有入侵企图的人得到，系统将受到攻击；

(4)即使是一个真正安全的系统，仍然无法防止系统的合法用户滥用权力；

(5)系统实施的访问控制策略的层次与系统的效率成反比关系，访问控制机制越严格，系统的效率就越低，可用性也越差。

相对来说，检测法显得更实用。我们可以从系统运行过程中产生的或系统所处理的各种数据中查找出威胁系统安全的因素，并对威胁做出相应的处理，这就是入侵检测(ID)，相应的软件或硬件称为入侵检测系统(IDS)。被检测的系统，即被保护的系统，称为目标系统。这里，对数据的检查可以同步于目标系统的运行，即检测是实时的，也可以滞后于目标系统。对威胁的处理称为响应，一般是采取一些预防性的措施，以阻止攻击的进一步发生，减小攻击对被检测系统造成的破坏，或者是保留攻击现场的相关数据，以作为受到攻击的证据。

近年来，入侵检测一直是很热门的话题。大量的关于入侵检测的文献都提出了自己的研究成果。但从总体上看，这些文献一般集中在对入侵检测方法的研究上，研究结果还不能有效地解决应用中的实际问题，如较高的漏报率或误报率。我们认为，对于入侵检测的研究，首要的是研究入侵行为的特点，对入侵行为进行总结和分类，然后据此确定应该获得用户行为的哪些相关数据以及如何获得这些数据，最后根据系统的安全漏洞和已获得的关于用户行为的背景知识找出攻击行为。不同结构的信息系统具有不同的优点或缺点，对系统的入侵行为也有着不同的特点，因此对入侵检测的要求也不会完全相同。本文旨在通过对入侵检测系统发展历史的研究，结合入侵检测的评价方法，说明入侵检测系统要针对不同结构的信息系统进行设计，才能达到比较满意的检测效果，并由此推断，在互联网迅速发展的时代，面向主机的入侵检测、面向网络的入侵检测以及面向应用程序的入侵检测结合起来形成的全面的入侵检测系统将是发展的必然趋势。

2. 入侵检测系统的发展历程

入侵检测系统从开始形成至今已有二十多年了，根据一些标志性事件的发生和标志性研究结果的出现，我们可以把

*)本文得到国家自然科学基金(No. 60073022)、中国科学院知识创新工程重大项目课题(KGCX 1-09)和中科院软件所培育基金项目(CXKE5143)等支持。

它的发展历史分为以下几个时期：(1)奠基时期(概念的形成和模型的建立)；(2)基于主机的入侵检测时期；(3)基于网络的入侵检测时期；(4)分布式入侵检测时期；(5)基于标准的入侵检测时期。

这种阶段划分方法以时间为线索，同时也体现了入侵检测系统在结构上适应被保护系统和满足安全需求的发展过程，便于看清IDS的发展趋势。

2.1 入侵检测概念的提出

1980年，James Anderson在为美国空军做的技术报告[1]中首次提出了入侵检测的概念，指出可以利用审计踪迹来检测对文件的非授权访问，并给出了一些基本术语的定义，包括威胁、攻击、渗透、脆弱性等。这可能是对入侵检测最早的研究。

该报告把渗透分为外部渗透和内部渗透两类，对应的入侵者分别称为计算机系统的外部入侵者和内部入侵者。报告根据攻击的形式把内部入侵者划分成“冒充者”、“合法用户”和“秘密用户”三类。报告中使用的检测方法是基于统计的，检测程序把某类会话(session)的资料，比如连接时间、输入输出的数据量、访问文件的数量等，同代表正常会话的一些阈值比较，当二者的差异超过既定的范围时，这次会话被当作异常。各类阈值是在对大量用户的类似行为作统计的基础上得出的平均值。

Anderson实现的审计是基于单个主机的，在应用软件层实现，其覆盖面不广，且完整性难以保证，这是与当时通过哑终端连接多用户主机的系统结构形式相适应的。但是，Anderson提出的关于计算机安全的一些基本概念、对攻击方式的分析和分类，为以后入侵检测系统的开发和检测技术的探索奠定了良好的基础。

2.2 入侵检测模型的建立

Anderson的基于审计进行入侵检测的思想是IDS系统的基础，对以后的IDS系统有着很大的影响，但这在当时并没有受到足够的重视。当时正处于计算机系统安全技术的奠基阶段，研究的重点在于访问控制和各种安全模型等基础性的安全保证措施。

1983年，美国国防部颁布了可信计算机系统评价标准[5]，简称TCSEC或橙皮书。该标准明确定义了各个安全级别(D级除外)的审计功能要求，从而推动了对依据审计数据加强系统安全的研究，使得安全系统成为有安全访问机制作基础、有审计和入侵检测作补充的完整的结构。

1986年8月，Dorothy E. Denning在她的论文[4]中第一次提出了入侵检测系统的抽象模型(称为IDES系统)，把入侵检测作为全新的安全措施加入到计算机系统安全保证体系中。该模型定义了入侵检测系统的基本组件：

- 主体：事件或活动的发起者，通常是用户；
- 客体：系统所管理的资源，比如文件、设备等；
- 审计记录：系统产生的关于主体操作客体的事件的记录；
- 活动简档：描述主体操作客体的行为特征，可以用统计数据或行为模式表示；
- 异常记录：检测到异常活动时产生的记录信息；
- 活动规则：在条件满足时所采取的动作，比如修改用户的活动简档、检测异常行为、把异常关联到可疑的入侵、产生报告等；
- 警报：发给系统管理员以使其注意的信息。

该模型不依赖于任何特定的系统和应用环境，也不假定被检测的攻击类型(如尝试攻击、冒充攻击、合法用户的渗透、病毒、拒绝服务等)，因此是一个通用的检测模型。在此之后开发的IDS系统基本都沿用了这个结构模型。可以用图1表示Denning提出的模型：

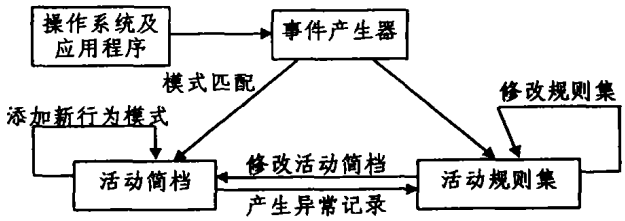

图1 通用IDS模型

该模型采用基于规则的模式匹配进行检测。审计记录产生时，要和活动简档中的记录进行匹配，依据活动简档中相匹配的记录中的活动规则决定下一步的动作。Denning的检测模型主要使用某个时间段内同类审计事件的数量、相关审计记录的时间间隔、某段时间内的系统资源使用情况等统计性测量参数，所采用的统计模型有操作模型、均值与方差、多元模型、马尔柯夫过程模型和时间序列模型等。

由于Denning的模型并不关心目标系统所采用的安全机制，因此也不检测攻击者针对已知的系统弱点进行攻击的特征行为。这可能是IDES系统的最大缺陷，事实上，针对已知的系统漏洞和攻击方法进行检测的准确率要更高些。

2.3 基于主机的入侵检测系统的繁荣

在Denning提出她的通用入侵检测模型后，很多机构先后推出了许多基于主机的入侵检测系统，这些系统有Haystack、MIDAS、IDES、Wisdom&Sence、NADIR等。此时，网络的普及程度还较低，网络上的攻击事件还很少见，网络安全尚未得到重视，所以这时推出的入侵检测系统大都是基于主机的。然而，基于主机的入侵检测对于整个信息系统的安全来说仍是不可或缺的，实际上，即使在基于网络的入侵检测技术成熟以后，仍有很多只针对单个主机的入侵检测系统出现。

Haystack原型[20]使用了两种检测方法：异常检测、基于入侵特征的检测。异常检测也分为两部分，即针对每个用户的异常检测和针对每个用户组的异常检测。基于入侵行为特征的检测方法弥补了Denning模型的不足，提高了入侵检测的准确率。两种检测方法的结合，克服了每种方法各自的缺陷，解决了入侵检测系统中的很多问题。

MIDAS[17]用于对美国国家计算机安全中心(NCSC)联网的大型主机进行入侵检测，它使用了启发式的入侵检测方法，形成一个入侵检测的专家系统。MIDAS的规则集分成两层，低层规则直接对特定类型的事件进行归纳，并将得到的可疑事件传递给高层规则集，以进一步判断是否需要向安全管理员告警。

1988年，SRI/CSL的Teresa Lunt等人对Denning的入侵检测模型进行了改进，提出了与系统平台无关的实时检测思想，并开发出了一个IDES系统[13]，用于检测对单一主机的入侵尝试。IDES的结构(如图2所示)与以前的系统相差较大，被检测的目标系统将自己产生的审计记录通过网络传递到运行IDES的另一台工作站，该工作站通过DBMS系统来管理审计记录数据库。然而，因为系统所分析的数据全部来自主机的审计数据，它仍然是基于主机的入侵检测系统。

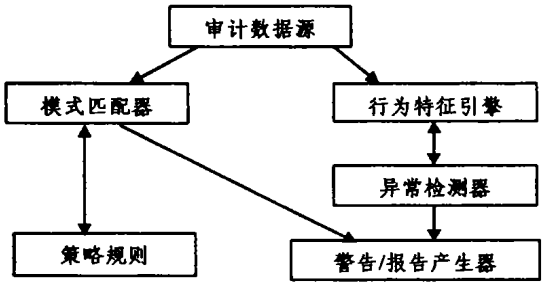

图2 IDES 结构框架

2.4 基于网络的入侵检测时期

随着计算机网络的快速发展，各种不同结构的计算机系统可以通过若干标准协议连接起来，信息系统的安全明显不再局限于单个主机的范围。1990年，加州大学 Davis 分校的 L. T. Heberlein 等人开发出了 NSM(Network Security Monitor)[7]，第一次直接把局域网上的网络信息流作为审计数据来源。通过局域网相连的若干异构主机有着不同的审计记录格式，把这些审计记录综合起来进行入侵检测可能会改善检测的效果。但要让入侵检测系统理解不同的审计记录格式就要对这些格式进行转换，而这将导致系统效率的下降。然而，局域网上的通讯是广播型的，并且使用像 TCP/IP、telnet、ftp 这样的标准协议，因此，NSM 就可以监听局域网上的所有网络通讯，并从中找出具有入侵企图的行为，而无需进行审计记录格式的转换，这不但改善了检测效果，也提高了效率。

NSM 的出现为入侵检测系统的发展史翻开了新的一页，自此形成了入侵检测系统的两个主要类别：基于网络的入侵检测和基于主机的入侵检测。

2.5 分布式入侵检测时期

基于网络的入侵检测与基于主机的入侵检测相比具有明显的优点，如部署数量少、能实时检测、具有操作系统独立性、对原有主机影响小，而且具有较强的隐蔽性，但同时也有较大的缺陷：只能检查一个广播型网段上的通信、难以处理加密的会话过程，对通讯协议的理解不全也使得基于网络的检测容易受到欺骗。由此看出，基于网络的入侵检测系统与基于主机的入侵检测系统各有优势，且具有互补性，把二者结合起来使用，有可能改善入侵检测系统的检测效果。这就是分布式入侵检测形成的原因。

1988年的 morris 蠕虫事件发生之后，网络安全才真正得到重视。美国空军、国家安全局(NSA)和能源部(DOE)共同资助空军密码支持中心(AFCSC)、Lawrence-Livermore 国家实验室、加州大学 Davis 分校、Haystack 实验室，开展对分布式入侵检测系统(DIDS)的研究，将基于主机和基于网络的检测方法集成到一起。1991年 DIDS 基本完成。

DIDS 系统[21,22]在框架上结合了 Haystack 系统[20]和 NSM 系统[7]的特点，进行数据的分布式监视、集中式分析。DIDS 采用三层结构，如图3所示。

DIDS 是入侵检测系统历史上一个里程碑式的产品。它通过收集、合并来自多个主机的审计数据和检查网络通讯，能够检测出多个主机发起的协同攻击。

由于网络的迅速发展，网上入侵事件的频发，基于网络的入侵检测成为研究的热点。而由于分布式入侵检测系统集成了主机检测和网络检测，而且偏向于检测复杂的分布式网络攻击，因此也常常和网络入侵检测合称为基于网络的入侵检测。

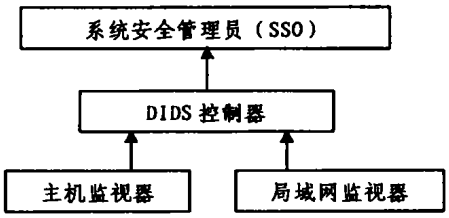

图3 DIDS 结构图

对于广泛意义上的网络入侵检测，它可以使用基于主机的入侵检测系统使用的各种检测方法。根据处理数据的不同方式，可以把基于网络的入侵检测系统分为集中式网络入侵检测系统和分散式网络入侵检测系统。集中式网络入侵检测系统就是把网络上分布的审计数据汇总起来集中进行分析，一般比较适合于小型网络，因为在大型网络中需要分析的审计数据是非常大的，中心节点的计算能力可能不足以应付。在分散式网络入侵检测系统中，数据收集是分布式的，数据分析也是分布式的。这类系统在结构上一般是分层的，易于扩展到较大规模的网络，因为数据分析是分布式进行的，各主机之间需要共享的审计数据较少。基于图的入侵检测系统 GrIDS[2,3] 和 SRI 公司开发的 EMERALD[16]都属于分散式网络入侵检测系统。

2.6 入侵检测系统的标准化时期

一个入侵检测系统往往只能适用于某种特定结构的信息系统环境，而且检测攻击的成功率也不会达到100%，然而随着应用程序的大型化、网络结构的复杂化以及入侵手法的多样化，单独使用一个入侵检测系统并不能有效地检测出入侵。可以猜想，把多个入侵检测系统结合起来，构成一个新的入侵检测系统，在更大的范围内和更高的层次上进行入侵检测，可能会更为有效。

从20世纪90年代到现在，入侵检测系统的研发呈现出百家争鸣的繁荣局面，并在智能化和分布式两个方向取得了长足的进展。随着越来越多的入侵检测产品的出现，自然地产生了把它们结合起来以最大可能地检测入侵的需求。为了使各个独立的入侵检测系统能够相互合作，就要求它们在信息格式、通讯协议等方面遵守共同的标准。为此，美国国防部高级研究计划署(DARPA)和互联网工程任务组(IETF)的入侵检测工作组(IDWG)分别发起制订了一系列建议草案，从体系结构、通信机制、语言格式、API 等方面来规范 IDS。1997年 DARPA 的 Teresa Lunt 等人提出制定公共入侵检测框架(CIDF)，现在已经完成。1999年6月，IDWG 就入侵检测也出台了一系列草案。目前这两个草案或建议还正处于逐步完善之中，它们为未来的国际标准提供了很好的素材。

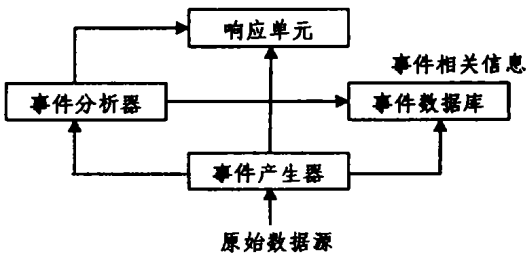

图4 CIDF 的入侵检测体系结构

CIDF[10]所做的工作主要包括四部分：IDS的体系结构、通信机制、描述语言和应用编程接口API。CIDF在IDES和NIDES的基础上提出了一个入侵检测系统的通用模型，将入侵检测系统分为四个基本组件：事件产生器、事件分析器、响应单元和事件数据库。很多IDS厂商都用数据收集部分、数据分析部分和控制台(console)部分三个术语来分别代替事件产生器、事件分析器和响应单元，有的还把数据收集部分与数据分析部分合称为传感器(Sensor)或探测器。组件是逻辑实体，一个组件可能是某台计算机上的一个进程甚至线程，也可能是多个计算机上的多个进程。组件之间以GIDO(统一入侵检测对象)格式进行数据交换。

入侵检测工作组(IDWG)[8]主要定义数据格式和交换规程，用于入侵检测与响应(IDR)系统之间或与需要交互的管理系统之间的信息共享。IDWG提出的建议草案包括三部分内容：入侵检测消息交换格式(IDMEF)、入侵检测交换协议(IDXP)以及隧道轮廓(Tunnel Profile)。IDMEF描述表示入侵检测系统输出信息的数据模型，IDXP是入侵检测实体之间交换数据的应用层协议，能够实现IDMEF消息、非结构文本和二进制数据之间的交换，并提供面向连接协议之上的双方认证、完整性和保密性等安全特征。隧道轮廓描述了对等体之间使用代理进行通讯时的IDXP交换。

以上两个建议或草案还没有正式成为标准，也没有一个入侵检测产品完全采用某个标准。但各种入侵检测系统的模型具有很大的相似性，只要各厂商按照统一的标准进行设计和实现，就能达到不同入侵检测系统之间的互操作。遵守国际标准的入侵检测系统是适应计算环境复杂化的必然要求。

3. 入侵检测技术

入侵检测系统在结构上的发展是与信息系统的结构变化密切相关的，但入侵检测的方式从开始至今却没有多大变化。入侵检测系统基本上使用两种检测方式，异常检测和误用检测。异常检测(Anomaly detection)就是抽取系统的静态形式和可接受的行为的特征，然后检测对静态形式的错误改动和可疑的动态行为。误用检测(Misuse detection)，又称特征检测(Signature-based detection)，它假设入侵活动可以用一种模式来表示，检测系统的目标是检查系统内发生的活动是否符合这些模式。

从入侵检测系统的发展历史来看，这两种检测方式很早就已得到使用了，很多系统已经把二者结合起来，以提高检测的准确程度。这两种检测方式各有短长，不可相互代替。近十几年来，还没有出现过与这两种方式有根本区别的其它检测方式，但用于误用检测的新模式却出现了不少。

可以把异常检测分为静态异常检测和动态异常检测两种。静态异常检测在检测前先保留一份系统静态部分的拷贝或特征表示，在检测时，如果发现系统的静态部分与以前保存的备份或特征之间有了偏差，则表明或者系统出了故障，或者系统受到了攻击。静态异常检测也可以说成是完整性检查。经常使用的特征表示方法有信息摘要、散列函数、校验和等。

动态异常检测的对象是用户行为或系统行为，这比检查静态部分的变化要困难得多。首先要创建活动简档文件来描述正常的行为，在检测时，把当前行为和活动简档中的正常行为进行比较，如果比较所得的差别值较大(超过了预定的标准差)，则判定发生了入侵。一般地，动态异常检测所采用的都是基于统计的方法，其最大的优点就是可以“学习”用户的使用习惯，从而具有较高的入侵检出率与可用性。但是它的“学习”能力也可能被入侵者利用，通过逐步“训练”入侵检测系统，可以使入侵事件变得符合“正常”操作的统计规律，从而避开入侵检测。

误用检测主要针对于使用已知的攻击技术进行攻击的情况。可以使用“入侵场景”(intrusion scenario)来确切地描述一个已知的入侵方式，一般地，这就是一个行为序列。当此行为序列完成时，就发生了一次入侵。早期的误用入侵检测系统(如MIDAS[17]系统)使用规则来描述所要寻找的入侵，规则的管理和使用要借助于专家系统，但因为没有按照入侵场景来组织规则，导致了大量规则的产生，而且这些规则难以解释和修改。为了克服这个缺点，以后的入侵检测系统使用了其它的入侵场景表示方法，其中包括基于模型的规则组织方法和基于状态转化的规则表示法，这样针对新发现的入侵方法时只需增加很少的规则。因为可以精确地表示入侵场景，误用入侵检测系统可以在入侵行为前进到一定程度时加以阻止。基于模型的误用检测和基于状态转化的误用检测都可以预见入侵的发生。

IDES[14]在开始时使用规则集描述入侵场景，但因为难于表达新的入侵场景，后来转向基于模型进行检测[6]，每一个入侵场景被分别建模，使得在对系统作修改时所需考虑的规则数大为减少。另外，由于只需触发检查入侵行为初始部分的规则，其它规则无需考虑，因而系统效率也有较大提高。

基于状态的入侵检测使用“属性-值”对来描述系统状态，状态之间的转化由动作来激发。入侵场景用状态转化图的形式描述。描述入侵场景的状态转化图由一个初始状态(入侵发生前)、一个或多个侵害状态(入侵完成后)和若干中间状态组成。在对三个基于状态转化的入侵检测系统USTAT[9]、NSTAT[11]和NetSTAT[24]作了分析、总结之后，G. Vigna等人重新设计了基于状态转化的入侵检测的工具框架[23]，其核心模块支持与应用域无关的入侵场景表示和入侵检测。这种新的设计扩展性强，易于移植到新的应用环境。

4. 入侵检测系统的评价

入侵检测系统已经成为计算机系统不可或缺的必要组成部分。然而入侵检测产品在结构和性能上有很大的区别，所使用的技术和适用的环境也不尽相同。为了能够比较不同的入侵检测系统，推动对入侵检测的研究，同时也为了帮助用户选择合适的入侵检测产品，我们有必要对入侵检测技术进行评价。1998年和1999年，美国国防部高级研究计划署(DARPA)分别进行了两次这样的评价[12]。当前，如何评价入侵检测系统也是研究的热点。

一般地，可以从以下几个方面去评价一个入侵检测系统：

(1)入侵检测系统的适应性。要考虑的因素有入侵检测系统所基于的操作系统、网络拓扑结构、所理解的网络协议、支持的网络应用等。没有一个入侵检测系统能够适应于所有的环境。

(2)能否保证自身的安全(健壮性)。由于自身功能的复杂性，入侵检测系统往往和其它系统一样存在安全漏洞。入侵检测系统首先必须保证自身的安全性。

(3)运行开销与维护开销的大小或难易程度，例如怎样更新入侵方式的特征库以检测出新出现的攻击方法。

(4)检测和报警的准确率。漏报(False Negative)是指在真正发生攻击时不能检测出攻击或不能做出响应，误报

(False Positive)是指在没有发生入侵的情况下将正常事件错误地当作入侵活动从而发出入侵警报。如果一个检测系统漏报率较高,则失去应用价值;而若误报率较高,则会给出过量的假警报,从而掩盖了真正的警报。一个实用的入侵检测系统应同时保持较低的漏报率和误报率。

(5)能检测的入侵类型数量。这种数量的计算应基于标准化的漏洞名称,如 MITRE 建立的通用漏洞纰漏(CVE)。

(6)入侵检测系统是否容易被躲避(Evasion)。入侵检测系统应能防止常用的躲避检测的方法,如:分片、异常 TCP 分段、TTL 欺骗、慢扫描、协同攻击等。

(7)系统的可伸缩性以及与其它系统的互操作性,包括系统能否方便地增加新的数据收集器、事件数据库的最大容量、系统各组件之间的通信带宽、对审计数据或日志溢出的处理等。随着 DARPA 和 IDWG 草案的完善,与其它系统的互操作性也应成为评价一个入侵检测系统的重要指标。

一个完备的入侵检测系统一定是基于主机和基于网络两种方式兼备的分布式系统,能够检测所有已知特征的入侵,并且能够识别一些新的攻击手段,又能保持较低的漏报率和误报率,同时具有相当的健壮性。除了这些功能上的要求外,在易用性上还要做到便于维护、升级,用户界面要友好,便于使用。因为入侵检测是很复杂的技术,入侵检测系统的人性化设计也是评价它的一项重要指标。当然,对于实际应用单位,价格是必须考虑的要点,入侵检测系统的性价比以及被保护系统的价值是更为重要的因素。

5. 入侵检测系统的发展方向

计算环境的复杂化,包括网络规模的迅速扩大和大型通用软件的普及,影响着入侵检测系统的发展方向。近年来,无论从规模上,还是从方法上,入侵(或攻击)行为都发生了变化,入侵的手段与技术也有了较大的改变。可以在互联网上找到现成的攻击工具,这使得攻击者的潜在数量大大增加。入侵者在实施入侵或攻击时往往同时采取多种入侵手段,而且往往从多个主机同时攻击目标(分布式攻击)。随着入侵检测系统的大量应用,入侵检测系统本身也成了被攻击的对象。

针对入侵和攻击技术的发展,今后入侵检测的发展方向主要有:

(1)分布式入侵检测,其中的关键技术为检测信息的协同处理与入侵攻击的全局信息的提取。

(2)面向应用的入侵检测。由应用程序解释的各种不同类型的数据(如加密过的数据),它们的语义只有在应用层才能理解,若想能够理解并分析这些数据,入侵检测系统必须面向应用层。现在已经有了如何进行应用程序级入侵检测的初步分析[18,19]。可以预见,应用层的入侵检测将成为新一代的入侵检测系统的特色。这样,入侵检测系统就成为一个以主机为中心、以网络为扩展面、以应用程序为高度的立体式的、全面的复杂系统。

(3)入侵检测系统之间的互操作性将大大增强,多个系统的相互配合,有利于应对攻击规模的扩大化,也有助于检测一些新型的攻击方式。入侵检测系统的相关标准草案将很快完善,并最终形成标准,为多系统的互操作提供基础。

然而,应该看到,入侵检测仅是保证信息系统安全的手段之一,无论检测技术多么有效,仍不能保证系统的绝对安全。必须把系统安全作为一个整体工程来处理,结合管理、数据加密、防火墙、病毒防护、入侵检测等多种技术、手段,全面评估受保护的系统,然后提出全面的安全解决方案。

结束语 入侵检测是保障计算机系统安全的一种手段。在二十几年的历史中,入侵检测系统的检测对象从主机的审计数据扩展到共享网段的网络通信,进而又扩展到广域网的分布式信息,并将向应用程序层渗透,形成对整个计算机系统全面的、立体式的安全检测。从最早 Denning 提出的通用模型到 DARPA 和 IDWG 提出的标准草案或建议,入侵检测系统在结构上已经比较成熟。今后,遵守共同的标准,相互间能够协作的入侵检测系统将成为必然趋势。在检测技术方面,基本上只有异常检测和误用检测两大类,但近来智能化的检测技术成为研究的热点,希望藉此能降低入侵检测的漏报率和误报率。

本文通过对入侵检测系统发展历史的研究,说明入侵检测系统要针对不同结构的信息系统进行设计,才能达到比较满意的检测效果。由此推断,在互联网迅速发展的时代,面向主机的入侵检测、面向网络的入侵检测以及面向应用程序的入侵检测结合起来形成的全面的入侵检测系统将是发展的必然趋势。

参 考 文 献

1 Anderson J P. Computer security threat monitoring and surveillance: [Technical report]. James P. Anderson Company, Fort Washington, Pennsylvania, April 1980

2 Staniford-Chen S, Cheung S, Crawford R, et al. GrIDS - A Graph Based Intrusion Detection System for Large Networks. In: 20th National Information System Security Conf. (NISSC), Oct. 1996

3 Cheung, Steven, Crawford R, et al. The Design of GrIDS: A Graph-Based Intrusion Detection System: [Technical report CSE-99-02]. The University of California, Davis Department of Computer Science. Jan. 1999

4 Denning D E. An intrusion-detection model. IEEE Transactions on Software Engineering, 1987, 13(2): 222～232

5 CSC-STD-001-83, Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. DoD Computer Security Center, Aug 1983

6 Garvey T D, Lunt T F. Model-Based Intrusion Detection. In: Proc. of the 14th National Computer Security Conf. Oct. 1991

7 Heberlein T, Dias G, Levitt K, et al. A network security monitor. In: Proc. of the 1990 IEEE Symposium on Research in Security and Privacy. IEEE Comput. Soc. Press, Los Alamitos, CA, USA, 1990. 296～304

8 http://www.ietf.org/html.charters/idwg-charter.html, April 4, 2002

9 Ilgun K. USTAT: A real-time intrusion detection system for UNIX. In: Proc. of the 1993 IEEE Symposium on Security and Privacy. Oakland, California, IEEE Computer Society Press, 1993. 16～28

10 Kahn C, Porras P, Stanford-Chen S, Tung B. A Common Intrusion Detection Framework. Submitted to Journal of Computer Security, July 1998

11 Kemmerer R. NSTAT: A Model-based Real-time Network Intrusion Detection System: [Technical Report TRCS-97-18]. Department of Computer Science, UC Santa Barbara, Nov. 1997

12 Lippmann R, et al. The 1999 DARPA Off-Line Intrusion Detection Evaluation. Draft of paper submitted to Computer Networks, In Press, 2000

(下转第155页)

参考文献

1 Bobbio A, Garg S, Gribaudo M, Horvath A, Sereno M, Telek M. Modeling Software Systems with Rejuvenation, Restoration and Checkpointing through Fliud Stochastic Petri Nets. In: Intl. Conf. on Petri Nets and Performance Models, PNPM99, Sep. 1999

2 Bobbio A, Sereno M, Anglano C. Fine Grained Software Degradation Models for Optimal Software Rejuvenation Policies. Performance Evaluation, 2001, 46: 45～62

3 Castelli V, et al. Proactive Management of Software Aging. IBM Journal of Research & Development, 2001, 45(2)

4 Dohi T, Goseva-Popstojanova K, Trivedi K S. Analysis of Software Cost Models with Rejuvenation. In: IEEE Intl. Symposium on High Assurance Systems Engineering, HASE 2000, Nov. 2000

5 Dohi T, Goseva-Popstojanova K, Trivedi K S. Statistical Non-Parametric Algorithms to Estimate the Optimal Software Rejuvenation Schedule. In: Pacific Rim Intl. Symposium on Dependable Computing, PRDC 2000, Dec. 2000

6 Garg S, Huang Y, Kintala C, Trivedi K S. Time and Load Based Software Rejuvenation: Policy, Evaluation and Optimality. In: First Fault Tolerance Symposium, FTS-95, Dec. 1995

7 Garg S, Huang Y, Kintala C, Trivedi K S. Minimizing Completion Time of a Program by Checkpointing and Rejuvenation, ACM SIGMETRICS 1996, May 1996

8 Garg S, Puliafito A, Telek M, Trivedi K S. Analysis of Software Rejuvenation using Markov Regenerative Stochastic Petri Nets. In: Int'l. Symp. on Software Reliability Engineering, ISSRE 1995, Oct. 1995

9 Garg S, Puliafito A, Telek M, Trivedi K S. On the Analysis of Software Rejuvenation Policies. In: Annual Conf. on Computer Assurance (COMPASS), June 1997

10 Garg S, Puliafito A, Telek M, Trivedi K S. Analysis of Preventive Maintenance in Transactions Based Software Systems. IEEE Transactions on Computers, Jan. 1998

11 Garg S, van Moorsel A, Vaidyanathan K, Trivedi K S. A Methodology for Detection and Estimation of Software Aging. In: Int'l. Symp. on Software Reliability Engineering, ISSRE 1998, Nov. 1998

12 Gray J, Reuter A. Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, San Mateo, CA, 1993

13 Huang Y, Kintala C, Kolettis N, Fulton N. Software Rejuvenation: Analysis, Module and Applications. In: IEEE Intl. Symposium on Fault Tolerant Computing, FTCS 25

14 Huang Y, Kintala C M R. Software Implemented Fault Tolerance: Technologies and Experience. In: Proc. of 23rd Intl. Symposium on Fault-Tolerant Computing, Toulouse, France, June 1993. 2～9

15 Li L, Vaidyanathan K, Trivedi K S. An Approach for Estimation of Software Aging in a Web Server. In: Intl. Symposium on Empirical Software Engineering, ISESE 2002, Nara, Japan, Oct. 2002

16 Marshall E. Fatal Error: How Patriot Overlooked a Scud, Science, Mar. 1992. 1347

17 Pfening A, Garg S, Puliafito A, Telek M, Trivedi K S. Optimal Software Rejuvenation for Tolerating Software Failures. Performance Evaluation, 1996, 27 & 28,

18 Trivedi K S, Vaidyanathan K, Goseva-Popstojanova K. Modeling and Analysis of Software Aging and Rejuvenation. In: IEEE Annual Simulation Symposium, April 2000

19 Vaidyanathan K, Harper R E, Hunter S W, Trivedi K S. Analysis and Implementation of Software Rejuvenation in Cluster Systems. ACM SIGMETRICS 2001/Performance 2001, June 2001

20 Vaidyanathan K, Selvamuthu D, Trivedi K S. Analysis of Inspection-Based Preventive Maintenance in Operational Software Systems. In: Intl. Symposium on Reliable Distributed Systems, SRDS 2002, Osaka, Japan, Oct. 2002

21 Vaidyanathan K, Trivedi K S. A Measurement-Based Model for Estimation of Software Aging in Operational Software Systems. In: Int'l. Symp. on Software Reliability Engineering, ISSRE 1999, Nov. 1999

22 林闯. 随机 Petri 网和系统性能评价. 清华大学出版社, 2000

(上接第49页)

13 Lunt T F, et al. IDES: The enhanced prototype, A real-time intrusion detection system: [Technical Report SRI Project 4185-010, SRI-CSL-88-12]. CSL SRI International, Computer Science Laboratory, SRI Intl. 333 Ravenswood Ave., Menlo Park, CA 94925-3493, USA, Oct. 1988

14 Lunt T, Jaganathan R, Lee R, Whitehurst A, Listgarten S. Knowledge-Based Intrusion Detection. In: Proc. of the 1989 AI Systems in Government Conf. March 1989

15 Miller B P, et al. A Re-examination of the Reliability of UNIX Utilities and Services: [Technical Report]. Department of Computer Science, University of Wisconsin, 1995

16 Porras P A, Neumann P G. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: 19th National Information System Security Conf. (NISSC), 1997. http://www.csl.sri.com/emerald/emerald-niss97.html

17 Sebring M M, et al. Expert systems in intrusion detection: A case study. In: Proc. of the 11th National Computer Security Conference, Baltimore, Maryland, NIST, 1988

18 Sielken R. Application Intrusion Detection. University of Virginia Computer Science: [Technical Report CS-99-17]. June 1999

19 Sielken R, Jones A. Application Intrusion Detection Systems: The Next Step. ACM Transactions on Information and System Security, Submitted 1999

20 Smaha S E. Haystack: An intrusion detection system. In: Proc. of the IEEE Fourth Aerospace Computer Security Applications Conf. Orlando, FL, USA, December 1988. IEEE, IEEE Computer Society Press, Los Alamitos, CA, USA

21 Snapp S R, Brentano J, Dias G V, et al. DIDS (Distributed Intrusion Detection System)- Motivation, Architecture, and An Early Prototype. In: Proc. of the 14th National Computer Security Conference, Oct. 1991

22 Snapp S R, Smaha S E, Teal D M, Grance T. The DIDS (distributed intrusion detection system) prototype. In: Proc. of the Summer USENIX Conf. San Antonio, Texas, USENIX Association, 1992. 227～233

23 Vigna G, Eckmann S T, Kemmerer R A. The STAT Tool Suite. In: Proc. of the 2000 DARPA Information Survivability Conf. and Exposition, 2000

24 Vigna G, Kemmerer R. NetSTAT: A Network-based Intrusion Detection Approach. In: Proc. of the 14th Annual Computer Security Application Conf. Scottsdale, Arizona, Dec. 1998