入侵异常检测研究综述

杨宏宇[1]，朱 丹[1]，谢 丰[2]，谢丽霞[1]

(1. 中国民航大学计算机科学与技术学院 天津 东丽区 300300；2. 中国信息安全测评中心技术研发部 北京 海淀区 100083)

【摘要】入侵检测是网络安全中极其重要的一环，异常检测是近年来入侵检测研究领域的热点。从分析入侵检测和网络安全模型间的关系开始，介绍入侵检测的概念和入侵检测系统的抽象模型，重点讨论基于网络数据、基于系统调用和基于系统调用参数的异常检测技术方法，对3种技术的重要研究方法进行了分析。指出入侵检测目前应尽量降低入侵检测系统对目标系统的性能影响和重点解决入侵异常检测系统的性能开销问题。随着网络环境的不断变化和入侵攻击手段的不断推陈出新，入侵异常检测未来的研究趋势之一是在入侵异常检测系统中增加可视化情景再现过程。

关 键 词 异常检测； 入侵； 网络数据； 系统调用； 系统调用参数

中图分类号 TP393.08 **文献标识码** A doi:10.3969/j.issn.1001-0548.2009.05.014

Survey of Anomaly Intrusion Detection Research

YANG Hong-yu[1], ZHU Dan[1], XIE Feng[2], and XIE Li-xia[1]

(1. School of Computer Science, Civil Aviation University of China Dongli Tianjin 300300;

2. Research and Development Department, China Information Technology Security Certification Center Haidian Beijing 100083)

Abstract Intrusion detection is an extremely important aspect of network security. The Anomaly intrusion detection research is one of highlighted topics of intrusion detection. The relationship between intrusion detection and network security model is reviewed. The concept of intrusion detection and the abstract model of intrusion detection system are introduced. Three developing technologies including network data based anomaly detection, system call based anomaly detection, and system call arguments based anomaly detection are discussed in detail. Most important research methods of those three technologies are summarized. Finally, the future development of this research domain is presented.

Key words anomaly detection; intrusion; network data; system call; system call arguments

随着网络技术的发展，信息安全的内涵也在不断延伸，从最初的信息保密性发展到信息的完整性、可用性、可控性和不可否认性，进而又发展到攻击、防范、检测、管理和评估等多方面的基础理论和实施技术。

由于计算机网络自身存在的局限性和信息系统的脆弱性，使得网络和计算机系统的硬件资源、通信资源、软件及信息资源等因可预见或不可预见，甚至是恶意的原因而遭到破坏、更改、泄露或功能失效，使信息系统处于异常状态，甚至引起系统的崩溃瘫痪，造成巨大的经济损失，以致保护网络中的信息免受各种攻击为根本目的的网络安全变得越来越重要。随着网络的发展，传统的计算机安全理论已不能适应动态变化的、多维互联的网络环境。

入侵检测(intrusion detection)是对入侵行为的发觉，通过对网络或计算机系统中的若干关键点收集信息并进行分析，从而发现网络或系统中是否有违反安全策略的行为和被攻击的迹象。入侵检测系统(intrusion detection system，IDS)就是按照一定的安全策略建立相应的安全辅助措施，发现入侵行为的软硬件组合保障系统。

对 IDS 的要求是：如果系统遭到攻击，IDS 应尽可能地检测到，甚至是实时地检测到入侵攻击，然后采取适当的处理措施。入侵检测系统作为安全技术其作用在于：识别入侵者和入侵行为；检测和监视已成功的安全突破；为对抗入侵及时提供重要信息以阻止事件的发生和事态的扩大。从这些角度看待安全问题，入侵检测技术是非常必要的，它将弥补传统安全保护措施的不足。目前，入侵检测研究已经成为信息安全领域的热点和难点。

众所周知，随着网络技术的发展和网络业务的增多，入侵攻击手段也总是不断推陈出新，也促使

收稿日期： 2009 – 01 – 29
基金项目：国家自然科学基金(60776807)；国家863计划重点课题(2006AA12A106)
作者简介：杨宏宇(1969 –)，男，博士，教授，主要从事网络与信息安全方面的研究.

入侵检测理论和技术的不断发展。本文主要对最近几年来入侵检测研究领域基于数据挖掘的异常检测、基于系统调用的异常检测和基于系统调用参数的异常检测等研究热点进行分析。

1 网络安全模型与入侵检测

1.1 网络安全模型

20 世纪 90 年代末，美国安氏公司(ISS)提出了自适应网络安全模型 P2DR[1]，成为目前国际上比较实际并可指导信息系统安全建设和安全运营的安全模型框架，如图 1 所示。P2DR 模型包含安全策略(policy)、防护(protection)、检测(detection)和响应(response) 4 个主要部分。防护、检测和响应组成了一个完整的、动态的安全循环，在安全策略的整体指导下保证信息系统的安全。

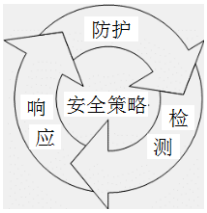


图 1 P2DR 模型

在该模型中，检测起着极其重要的纽带作用，具体包括以下几点：

(1) 检测是静态防护转化为动态防护的关键。只有充分了解当前的安全态势，才能更加明确防护的重点以及防护的漏洞。

(2) 检测是动态响应的依据，系统是否能够采取合适的响应方式关键是看检测的结果是否准确。

(3) 检测是落实或强制执行安全策略的有力工具，安全策略的建立不应该是静态不变的，而必须根据系统当前状况动态调整以适应安全需要，而系统当前安全状况的信息主要来自检测系统。

1.2 入侵检测

国际上对入侵检测的研究开始于 20 世纪 80 年代，文献[2]首次引入入侵检测的概念，定义“入侵检测就是发现入侵企图或者潜在的恶意操作的技术，这些操作可能会导致非认证存取、操纵信息或导致系统不可靠和不可用”。此后，入侵检测技术得到了广泛的研究。

入侵检测方法从计算机网络系统中的若干关键点收集并分析监测信息，判断网络中是否有违反安全策略的行为和遭到袭击的迹象。它能在不影响网络性能的情况下对网络进行监测，为系统提供对内部攻击、外部攻击和误操作的有效保护。另外还可以提供相应的防护手段，如记录入侵证据以便跟踪、恢复和断开网络连接等，因此入侵检测是网络安全中极其重要的组成部分。

文献[3]提出了入侵检测系统的抽象模型，首次将入侵检测作为一种计算机系统安全防御措施提出，成为入侵检测发展史上的里程碑。由于该模型与具体系统和具体输入无关，因此对此后的很多实用系统都有借鉴意义，至今仍在入侵检测中得到广泛应用。该模型包含事件产生器(event generator)、活动记录(activity profile)和规则集(rule set) 3 个主要部件，如图 2 所示。事件产生器负责为模型产生事件，可以根据具体应用环境而有所不同，一般情况下可以从审计记录、网络数据包以及其他可观察行为中提取事件。活动记录是整个检测系统的核心，用于保存目标的特征或者正常模式。规则集包括一个检测引擎和相应的动作集合。当观察模式出现异常状况时，活动记录产生异常记录报告，规则集对异常记录报告进行检测并产生相应响应。此外，反馈也是模型的一个重要组成部分，现有的事件会引发系统的规则学习，加入新的规则或者修改已有规则。

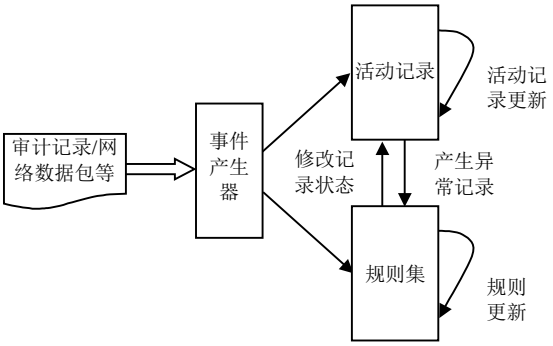


图 2 入侵检测系统抽象模型

如果从技术上分类，可将入侵检测分为误用检测(misuse detection)和异常检测(anomaly detection)两类。

误用检测也被称为特征检测(signature-based detection)，指运用已知攻击方法，通过判断已定义好的入侵模式是否出现进行检测[4]。该方法由于依据具体特征库进行判断，所以检测准确率很高，非常类似于现在的病毒检测，目前的商用产品主要采用这种方法。但是误用检测存在一个很大的弊端，就是它只能检测出那些包含在特征库里的已知入侵行为，而不能检测那些新出现的攻击或者已有攻击的变种。当一种新的攻击出现时，由于特征库中没有该攻击的特征，因此系统并不能检测出该种入侵。只有当安全专家人工分析这种攻击，并找出它独有

的特征加入到特征库中时，系统才具备检测能力。因此其缺点是显而易见的:(1) 从攻击发生到特征提取并更新特征库需要一段时间，而在此期间攻击可能已经造成很大的损失。(2) 通常需要人工提取特征，耗费大量的人力。

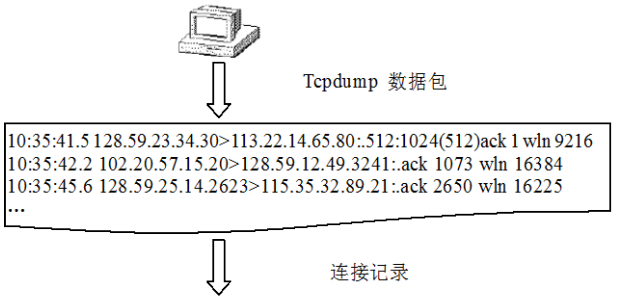
目前，异常检测研究尚处于初期探索阶段。异常检测根据使用者的行为或资源使用状况判断是否发生入侵，而不依赖于具体攻击是否出现。其优点是可以发现一些未知的攻击模式，但是该方法误报率较高，对于其过程和方法既没有统一的结论，也没有可以投入使用的完整系统。

2 基于网络数据的异常检测

对网络数据的异常检测分为基于网络数据包的检测和基于网络连接记录的检测[5]两种。前者主要是检查每一个网络数据包，分析包的协议字段甚至负载内容，检查是否有违背协议或者异常的负载内容。它的优点是可以从报文级实时发现入侵，然后进行过滤，但同时也导致处理数据量的急剧增加，因此在高速数据流中常常发生丢包现象。更重要的是，由于单个数据包的内容信息太少，很难直接用于异常检测，并且各包之间没有关联，完全是独立分析的。

基于网络数据的异常检测方法将数据还原成基于传输层的连接记录后，就具有多方面的特征，包括会话期间的内容特征，如传送的字节数、建立的文件数、口令尝试的次数、错误分片的次数、尝试 su 命令的次数等等，这些信息对于区分连接的正常与否十分重要。通常情况下正常连接很少反复猜测口令，因此这个属性的取值应该是很小的。相反对于恶意的口令猜测，这个属性值应该比较大。通过对这些取值进行建模，往往可以发现异常连接。

文献[6-10]使用数据挖掘技术对网络审计数据进行分析，它们的研究首先将网络数据包还原成基于传输层的连接记录(session record)或会话记录，然后从连接(会话)记录中选出部分属性作为特征，每一个连接记录用一个特征向量表示，最后利用分类器对这些特征向量进行分类。因为训练的数据都是正常数据并只有一个类别(即正常类别)，为了能进行分类，使用网络服务(service)端口作为网络连接记录的类别，根据大量的正常连接记录生成分类模型。在检测过程中，根据分类模型对当前的连接记录进行分类，并与实际服务类型进行比较。如果出现大量的分类错误，就可以断定出现了某种异常。这种方法实际上是假定异常行为与正常行为具有很大差别，因此基于正常行为的分类器难以正确识别异常行为，从而导致大量的分类错误。另外，该研究利用关联算法挖掘出大量的频繁模式(frequent episode)，最终实现了一个原型系统 madam-id，并参加了 DARPA 1998 测试，取得了较好的效果[11-12]。该模型的预处理框架(即网络连接记录的提取过程)如图 3 所示。



time	dur	STC	dst	bytes	srv	rlang	…
10:35:39.1	5.2	A	B	42	http	SF	…
10:35:40.4	20.5	C	D	22	uscr	REJ	…
10:35:41.2	10.2	E	F	1 036	ftp	SF	…
…	…	…	…	…	…	…	…

图 3 网络数据预处理框架

文献[13-16]在审计数据分析系统 ADAM 中同样使用关联规则和分类方法处理网络数据，但是与文献[6-10]所使用的方法有所不同。该方法首先将训练数据分为两部分：

(1) 完全正常的网络数据，对这些数据采用关联算法进行处理，从中提取出频繁集构成网络正常行为的模式(profile)。

(2) 已经标注了类别的网络数据，利用这些数据训练一个决策树分类器。检测时，如果数据包含 profile 中的内容，则认为是正常数据；否则就认为是可疑数据，然后对可疑数据采用已经训练好的分类器进行进一步识别。

因为对数据集进行类别标注比较费时费力，因此许多研究人员开始探索使用聚类方法发现网络数据中的异常点。该方法假设正常数据与异常数据是明显可分的，而且正常数据往往聚成若干个簇(cluster)，异常数据表现成一些离散的点，也称为离异点(outlier)，就可以利用数据挖掘中的离异点检测方法进行检测。文献[17-21]设计的 MINDS 系统就是一个利用 LOF (local outlier factor)算法发现异常数据的原型系统。MINDS 系统从每一个网络连接中提取 8 个内容属性和 8 个统计属性作为特征。所谓

内容属性是指每一个连接自身的属性，如源地址、源端口、目的地址、目的端口等等。统计属性是指多个相关连接的统计特征，如一个时间窗口内同一个源地址对应的不同目的地址的数量等等。以这 16 个特征作为特征向量，再利用 LOF 算法计算出每一个连接的异常得分值，按照得分高低进行排序，最后组织成检测报告提交给网络管理员。系统采用真实的网络数据进行试验，并与基于特征的检测系统 snort 进行比较。试验结果表明，MINDS 系统检测出了许多新的入侵行为(snort 并不包含这些入侵的特征)。

文献[22-23]将聚类方法用于网络连接数据，其研究目标是使用无监督方法学习网络的正常行为模式，也就是训练集不再进行标注。为了防止大量异常数据聚集成“大团”，首先对训练数据进行预处理，使之只保留少量的异常数据(异常数据量小于总数据量的1.5%)。对数据聚类之后，将那些超过一定规模的聚类作为正常模式(profile)。当检测一个新连接记录时，根据该连接记录与正常模式的相似性判断其是否异常。文献[24]采用类似的思路进行试验，不同之处在于其采用基于密度和网格的聚类方法发现异常数据。文献[25]主要关注自适应学习问题以适应网络的动态变化。文献[26-28]则在检测中考虑了代价，并加入人工异常增强系统的检测性能，该研究表明加入人工异常有助于提高系统的检测率。

3 基于系统调用的异常检测

大多数操作系统至少被划分为内核层和应用层两个层次。内核为应用程序提供最基本的系统服务，比如 I/O 管理、虚拟内存、任务调度等等。这些功能以“系统调用”方式提供给用户。应用程序可以在自己代码中使用系统调用，来实现对系统的访问或对内核功能的调用。系统调用是应用层使用内核层功能的唯一接口，在应用程序代码中，看上去就像一般的库函数一样，只是在编译连接时，系统调用不像库函数那样把自己的代码嵌入到应用程序的可执行代码中。

许多操作系统的内核不以进程方式工作。应用程序进程通过系统调用请求内核的服务。系统调用的代码在内核态执行，并在系统调用返回后才继续执行应用程序的代码。在一个进程进行了系统调用，等待系统调用返回时，它将暂时不参加进程调度，只有当它们从内核服务退出(既系统调用返回)后，进程才被调度程序调度。

通过研究发现，不同系统调用对系统的危险程度是不一样的。根据REMUS的划分[29]，系统调用按照功能可以分为文件系统、进程管理，模块管理、内存管理、时间操作、通信、系统信息等，如表1所示。每个功能类又根据其对系统可能的危险程度分为允许完全控制系统、可用于DoS攻击、可用于改变或激活进程和没有危险4个级别，并将危险程度越小的系统调用赋予一个较大的危险级别号。这种划分对应了一个危险层次，即若$m \geq n$，则危险级别为n的系统调用也可能用于完成危险级别为m的攻击。

表1 基于功能的系统调用分类

编 号	1	2	3	4	5	6	7	8	9
功能类别	文件系统	进程管理	模块管理	内存管理	时间操作	通 信	系统信息	保留未用	还未实现

在设计操作系统时，哪些功能放入内核是由设计者决定的。系统运行时，内核代码和应用程序代码的分界是由硬件提供的保护机制决定的，内核在应用进程不可访问的地址空间中运行。一些特权指令，只有在内核态下才可以被执行，可保证内核的安全。系统调用通常通过一个硬件自陷指令(trap)实现，它改变 CPU 的执行方式和地址空间的映射，进入内核态。

通常网络入侵和系统入侵的目标是为了获取敏感信息，如访问、修改某些重要文件，甚至控制主机执行新的远程攻击，这些操作大部分都需要内核层提供的功能，因此就不可避免地要使用多个系统调用。通过监视系统调用的执行，可以在很大程度上发现入侵行为。相比监视普通库函数而言，因普通库函数只是在应用层上执行了一些操作，不涉及内核提供的核心功能，另外普通库函数远远多于系统调用，导致监视普通库函数过于复杂，而且容易产生大量的误报，故目前的研究主要集中在系统调用的分析上。

文献[30-32]从免疫系统的角度考虑计算机系统的保护机制。免疫系统最基本和最重要的能力是识别“自我/非自我”(self/nonself)，换句话讲，它能够识别哪些组织是属于正常机体的，不属于正常的就认为是异常，这个概念和异常检测的概念相似。生

物免疫系统使用氨基酸、蛋白质碎片来完成“自我”的分辨任务。通过大量试验发现：对一个特定的程序来说，其系统调用序列是相当稳定的，这些系统调用序列可以作为程序的“self”。使用短序列匹配算法(sequence time-delay embedding, STIDE)，用于计算实际系统调用序列与正常序列模式的相似程度。在训练阶段，收集程序正常运行时的所有短序列(即 *N*-Gram，*N* 可以取不同值)，并将这些短序列放在数据库中。在检测阶段，若发现一个短序列不在数据库中，则认为该短序列是异常序列。当异常序列累加到一定值时，根据一定的方法判断是否发生入侵。试验表明该方法是有效的，系统可检测出许多类型的攻击。

文献[7-8]使用机器学习方法来检测系统调用，分别采用两种方式进行试验：

(1) 利用规则学习算法 RIPPER 学习正常系统调用短序列和异常短序列的特点。该方法需要有一个包含正常短序列和攻击短序列的训练集(其中每一个短序列都已经标注类别)。在训练过程中，产生多条 if-then 类型的规则表明短序列的特点。如存在这样一个规则：“if P_2=104 and P_7=112 then sequence is normal”，其中 $P_i=j$ 表示在给定序列中第 i 个系统调用是 j(其中的 104 和 112 都是系统调用 ID)。这条规则表示：若一个短序列的第二个系统调用是 104，第 7 个系统调用是 112，则这个短序列是正常类型。该方法使用 Forrest 的数据集进行试验并取得一定效果。但是，这种方法需要标注短序列的类别，实际上属于误用检测，并不能发现新的攻击模式。表 2 显示了该研究提供的标注类别的系统调用序列的一个范例。

表 2 系统调用序列的类别标注

系统调用序列(N=7)	类别
4 2 66 66 4 138 66	正常
⋮	⋮
5 5 5 4 59 105 104	异常

(2) 利用 RIPPER 产生的规则进行预测，最后根据预测错误率判断是否产生入侵。在该方法中，将 *N*-Gram 看作是一个待分类的特征向量，前 $N-1$ 个调用作为该向量的特征，第 N 个调用作为该向量的类别，然后利用 RIPPER 建立一系列规则，表明“给定前 $N-1$ 系统调用，则最后一个系统调用应该是多少”。由于该方法不需要攻击短序列，因此可以用于异常检测。

有许多研究人员使用多种建模方式分析系统调用短序列，包括 SVM[33]、隐马尔可夫模型[32, 34-39]、神经网络方法[40]、有限状态自动机[41-44]和数据挖掘[45]等等。

文献[46-48]认为使用不定长的滑动窗口划分系统调用序列更能反映程序的行为，并通过 Teiresias 算法得到不同长度而且不重叠的系统调用短序列，然后利用模式匹配发现异常。试验结果表明该方法比固定窗口划分在检测准确性上要高一些，但会耗费更多时间。文献[49]利用信息论和概率模型选择不同长度的窗口，文献[50]则从数据压缩的角度分析变长窗口的划分。

文献[51]进一步考察了检测到入侵后如何采取措施阻断入侵行为。将检测模块植入到 Linux 内核中检查每一个系统调用。当发现可疑行为时，根据概率值分别采用延迟或者阻断措施。该方法的主要缺点是会降低系统的性能。

但是，目前的部分研究并没有从序列的角度分析系统调用，而仅仅只考虑每一个系统调用出现的频率。因为在文本分类领域中，常常只计算每一个 term(词项)的出现频率而不考虑 term 之间的关系，所以文献[52]采用类似的思想分析系统调用。该方法将每一个进程迹看作是一篇文档并用一个向量表示，而向量的每一维是不同的系统调用，其值为该系统调用在进程迹中出现的频率，检测时使用 KNN 分类器对目标进程进行判断。文献[53-54]采用类似的方法，将所有系统调用看作 term，用词袋模型(bag of term)构建分类器，最后通过分类器判断是否发生攻击。但是这种基于频率的方法存在一个很大的缺点：由于只考虑了每一个系统调用出现的频率，因此入侵者很容易躲过检测，如可以通过少量操作来获取敏感信息但并没有引起频率的巨大变化，或者通过加入一些无关指令来保持频率的一致。

与 Linux/UNIX 系统不同，Windows 系统没有明确的“系统调用”概念，但也存在着一个应用层和内核层交互的接口，称为“Native API”。一些研究人员提出通过监视 Native API 发现基于 Windows 系统的入侵行为[55-56]。该方法最主要的特点是分析 Native API 而不是系统调用，因而可以监控基于 Windows 平台的个人主机。该方法采用类似于前面的建模方法，着重关注 API 调用的序列关系。

前面的研究主要是从检测方面分析系统调用，而有些研究人员从另一方面分析这种方法。因为很多系统都仅仅根据系统调用名检查是否发生入侵，

文献[57]详细分析了该种方法的弊端，并提出使用“模仿攻击”(mimicry attack)躲避这一类检测系统。该方法基于两个前提：(1) 攻击者知道检测系统的检测机制；(2) 攻击者通过观察程序的运行，可以得到类似于检测系统使用的正常模式数据库。显然，这两个条件是可以满足的。在此基础上，攻击者可以采用多种方式，包括加入“无关”系统调用(no-ops)到调用序列中。这些“无关”调用就是指那些不会产生恶意影响、与攻击目标无关的系统调用，其目的就是为了将异常序列伪装成正常序列。试验结果表明，采用这些伪装方法之后，可以成功地躲避短序列匹配方法的检测。文献[58]则采用静态分析方法构造模仿攻击。

上述研究主要集中在采用哪一种模型来更好地表示程序的运行行为。它们的处理对象是系统调用名字，并没有考虑系统调用的其他信息(如参数和返回值)，同时也不区分同一调用在不同位置的调用情况。

显然，不同位置的系统调用应该是不一样的，即使它们具有同一个调用名。为了描述方便，可以用 open@L_1 表示在 L_1 这个位置上调用 open。由于 open@L_1 已经包含了位置信息，则 open@L_1 应该与 open@L_2 是有区别的，因为它们调用时的位置不一样。但是如果仅仅采用系统调用名，就没有办法区分它们之间的不同。因此有必要研究一种更高效的方法识别不同位置的系统调用，其优点应该是：

(1) 可以更加细粒度地区分每一个系统调用。

(2) 能够有效地抵抗模仿攻击，因为攻击者不仅仅需要伪造系统调用名，还需要伪造调用时的地址，增加难度。

(3) 识别出不同位置的系统调用，有利于以后的参数分析。

4 基于系统调用参数的异常检测

基于序列的检测方法存在的不足是只能提供“是否发生入侵”这样一个结论性判断，不能提供更多可读的、有意义的信息。如系统报告的下列序列是异常序列：

$$\text{open@}L_1\rightarrow\text{read@}L_2\rightarrow\text{mmap@}L_3\rightarrow\text{write@}L_4\rightarrow\text{write@}L_5$$

显然，网络管理员很难从这样的信息中获取有价值的内容，只能根据系统给出的结论性判断做出相应对策，这是远远不够的。人们希望入侵检测系统能够提供更多易于理解的、有价值的内容以方便网络管理员更好地理解到底发生了什么事情。这样做有两个明显的好处：

(1) 由于目前的技术限制，异常检测系统总会存在一些误报，如果系统在报警的同时能提供更多可读信息，那么就可以方便网络管理员判断这是误报还是真的入侵，从而有利于降低由于误报而导致的不必要损失。

(2) 网络管理员可以从这些附加信息中获取更多的关于入侵的知识，比如入侵修改了哪些文件、向哪些地址发送了数据、或者启动了哪些从来没有使用过的程序，这些知识可以指导网络管理员做出相应的修补工作。

另外，基于序列的检测方法是从控制流的角度分析入侵，它通过监视控制流的变化来判断系统是否受到攻击。对于目前的很多攻击，这样的检测已足够，因为攻击者往往通过执行恶意动作达到攻击目的，而这些恶意动作往往与正常的操作序列不同[59]。但是非控制数据攻击(non-control-data attacks)[60]、条件竞争攻击(race condition attacks)[61]等，仅仅依靠序列关系很难被检测出来。正如前面模仿攻击(mimicry attack)中提到的那样，攻击者也能了解目前的检测手段，会不断地改变攻击策略。尽管改变一个攻击策略需要很大的代价(如耗费更多的时间、需要更多的技术)，但一旦攻击成功，就能获得更多的好处，包括获取密码等敏感信息。国家计算机网络应急技术处理协调中心报告指出，目前网络攻击更偏向于对特定用户群体的信息窃取，促使系统的安全保护应尽可能地深入。

非控制流攻击的程序片断首先读取 passwd 文件和 shadow 文件进行身份验证，然后读写一个临时文件。假定上述对 buf 的操作过程中没有产生新的系统调用，则得到的序列关系为：

$$\text{open@}L_1\rightarrow\text{read@}L_2\rightarrow\text{close@}L_3\rightarrow\text{open@}L_4\rightarrow\text{read@}L_5\rightarrow\text{close@}L_6\rightarrow\text{open@}L_7\rightarrow\text{write@}L_8\rightarrow\text{close@}L_9$$

若在read@L_5与close@L_6之间发生溢出，破坏了open@L_7第一个参数的内容，则open@L_7将以读写方式打开任意文件，而序列依然保持不变并被判断为正常，也就是说，仅仅通过检查序列关系不能发现入侵行为。相反，如果对open@L_7的参数进行检查，就会很容易地发现此刻入侵试图读写一个敏感文件，从而发生报警。

目前，基于系统调用参数的入侵检测研究不是很多。文献[62-64]通过学习每一个系统调用参数的长度、字符分布、字符结构等信息，为每一个系统

调用建立一个模型，并利用这些模型判断是否发生异常。该方法主要关注每一个调用发生时的参数信息，没有考虑不同调用之间参数的关联，也没有区分不同位置的调用。如，对于系统调用 open，不同位置的 open 都使用同一个检测模型。另外，文献[65-66]将同样的方法应用到 web 检测器中，检查输入的 URL 是否异常。

文献[67-69]使用规则学习算法 LERAD 从训练集中学习包括序列和参数的规则，最后利用规则进行判断。

文献[70]也注意到在检测过程中，仅仅给出简单的异常报告是不够的，应该给出更多的参数信息。该文献的方法借助文法分析思路，利用堆栈保存每一个系统调用的参数(称为语义分析栈)，但检测方法依然采用序列方式，只是在发现异常序列后，不仅输出异常的系统调用序列，还输出更多的附加信息，包括用户输入请求、语义分析栈保存的系统调用参数等信息。因此，本质上，该方法在检测时依靠系统调用的序列，但在输出结果时包含参数信息，只是采用一个栈保存调用时的参数，应该说该方法也难以防止非控制流攻击。

非控制流攻击实例如下：

…

L_1: fd1 = open(“/etc/passwd”,O_RDONLY);

L_2: nsize = read(fd1,buf,sizeof(buf)));

…/*根据读取的 buf 内容进行认证。假定这个过程没有产生新的系统调用*/

L_3: close(fd1);

L_4: fd1 = open(“/etc/shadow”,O_RDONLY);

L_5: nsize = read(fd1,buf,sizeof(buf)));

…/*操作读取的buf内容，假定这个过程没有产生新的系统调用*/

L_6: close(fd1);

L_7: fd1 = open(“/tmp/tmp.out”,O_RDWR);

L_8: write(fd1,buf,nsize);

L_9: close(fd1)。

文献[71]提出利用参数学习来检测这些基于数据流的攻击。将参数学习分为一元参数学习和二元参数学习两种。一元参数学习关注每一个系统调用的参数信息，二元参数学习关注不同系统调用之间的参数关系。为了能够进行有效的学习，参数学习方法事先定义了一些关系，然后每分析一个系统调用时，依次与前面第 K 次出现的系统调用进行比较，看它们之间的每一对参数是否满足这些关系。目前该研究只实现了 $K=1$ 的情况。

下面通过一个实例描述参数学习方法的分析过程。假定 X、Y、Z 都表示系统调用，为了描述方便，给每一个调用赋予一个下标值。用 Y_1 表示下标为 1 的 Y，即第 1 个 Y，Y_4 表示下标为 4 的 Y，即第 2 个 Y。Y=1 表示系统调用 Y 的参数值取为 1。为了描述方便，本例只考虑参数为整数的情况，即：

$$\underset{1}{Y=1,Z=2,}\ \underset{2}{X=1,}\ \underset{3}{Y=2,}\ \underset{4}{X=2}\ \underset{5}{}$$

因为系统调用按照一定顺序依次发生，所以学习过程也按下列步骤依次进行：

(1) 首先分析 Z_2 的参数与 Y_1 的参数是否满足预定义的关系；

(2) 然后分析 X_3 的参数与 Z_2 和 Y_1 的参数是否满足预定义的关系；

(3) 接着分析 Y_4 的参数与 X_3、Z_2 和 Y_1 的参数是否满足预定义的关系；

(4) 最后分析 X_5 的参数与 Y_4、X_3、Z_2 的参数是否满足预定义的关系。

步聚(4)中，X_5 不再与 Y_1 的参数进行比较，因为 Y_1 是 X_5 前面的第 2 次出现(最近一次出现是 Y_4)。根据上述分析过程，最后可以得到下列关系：X 的取值总是等于前面第 1 次的 Y 的取值(如 X_3 等于 Y_1，X_5 等于 Y_4)。

文献[71]的学习方法的优点是除了人工定义一些关系外，不再需要领域知识(因为系统会不断地重复比较是否满足这些关系)，但缺点是需要耗费大量时间，并且得到的部分关系只是形式上的满足，并不具有必然原因。

5 结 束 语

本文从分析入侵检测和网络安全模型间的关系开始，介绍入侵检测的概念和模型，重点探讨了基于网络数据、基于系统调用、基于系统调用参数的异常检测技术，并就每种技术的最新研究方法的优劣作了较深入的比较分析。

入侵异常检测还有许多亟待解决的问题，如何尽量降低入侵检测系统对目标系统的性能影响。目前影响目标系统性能最大的环节在于获取目标程序的系统调用迹，应重点解决入侵异常检测系统的性能开销问题。

随着网络环境的不断变化和入侵攻击手段的不断推陈出新，入侵检测也需要不断地发展。入侵异常检测未来的研究趋势之一，是在入侵异常检测系

统中增加可视化情景再现过程，不仅提供结论信息，更提供完整的入侵过程，从而有利于提高人们对入侵过程的认识，增强网络防范。

参 考 文 献

[1] Information Security One (China) Ltd. Security service ideas and standard[EB/OL]. [2009-03-25]. http://bj.is-one.net/safe/standard/P2DR/2008, 12.

[2] ANDERSON J P. Computer security threat monitoring and surveillance[R]. USA, 1980.

[3] DENNING D E. An intrusion detection model[J]. IEEE Transactions on Software Engineering, 1987, 13(2): 222-232.

[4] CAULKINS D, LEE J, WANG M. Packet-vs. session-based modeling for intrusion detection systems[C]//Proc of the International Conference on Information Technology: Coding and Computing (ITCC 2005). Las Vegas, Nevada, USA: IEEE Computer Society, 2005: 80-87.

[5] 张世永. 网络安全原理与应用[M]. 北京: 科学出版社, 2003.
ZHANG Shi-yong. Principle and application of network security[M]. Beijing: Science Press, 2003.

[6] LEE W, STOLFO S J, CHAN P K. Learning patterns from Unix process execution traces for intrusion detection [C]//AAAI Workshop of AI Approaches to Fraud Detection and Risk Management. Menlo Park, CA: AAAI, 1997: 50-56.

[7] LEE W, STOLFO S J. Data mining approaches for intrusion detection[C]//Proc of 7th USENIX Security Symposium. Berkeley, CA, USA: USENIX Association, 1998, 7: 6-16.

[8] LEE W, STOLFO S J, MOK K W. W. Mining audit data to build intrusion detection models[C]//Proc of the 4th International Conference on Knowledge Discovery and Data Mining (KDD). New York, USA: AAAI Press, 1998: 66-72.

[9] LEE W, STOLFO S J, MOK K W. A data mining framework for building intrusion detection models[C]//Proc of the IEEE Symposium on Security and Privacy (ISP). Oakland, California, USA: IEEE Computer Society, 1999: 120-132.

[10] LEE W. A data mining framework for constructing features and models for intrusion detection systems[D]. New York: Columbia University, 1999.

[11] LIPPMANN R, CUNNINGHAM R K, FRIED D. Results of the DARPA 1998 offline intrusion detection evaluation[C]//Proc of Recent Advances in Intrusion Detection (RAID). West Lafayette, Indiana, USA: Springer-Verlag, 1999: 162-168.

[12] LIPPMANN R, FRIED D, GRAF I, et al. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation[C]//Proc of the DARPA Information Survivability Conference and Exposition (DISCEX). Hilton Head, South Carolina, USA: IEEE Press, 2000: 12-26.

[13] BARBARA D, COUTO J, JAJODIA S, et al. ADAM: A tested for exploring the use of data mining in intrusion detection[J]. SIGMOD Record, 2001, 30: 15-24.

[14] BARBARA D, WU N, COUTO J, et al. Detecting intrusions by data mining[C]//Proc of 2nd Annual IEEE Systems, Man, and Cybernetcis Information Assurance Workshop. West Point, NY, USA: IEEE Computer Society, 2001: 11-16.

[15] BARBARA D, WU N, JAJODIA S. Detecting novel network intrusion using Bayes Estimators[C]//Proc of the First SIAM Conference on Data Mining. Chicago, IL, USA: IEEE Computer Society, 2001: 135-141.

[16] LI Y, WU N, JAJODIA S, et al. Enhancing profiles for anomaly detection using time granularities[J]. Journal of Computer Security. 2002, 10(1-2): 137-157.

[17] DOKAS P, ERTOZ L, KUMAR V, et al. Data mining for network intrusion detection[C]//Proc of NSF Workshop on Next Generation Data Mining. Baltimore, Maryland, USA: AAAI/MIT Press, 2002: 21-30.

[18] ERTOZ L, EILERTSON E, LAZAREVIC A. Detection of novel network attacks using data mining[C]//Proc of Workshop on Data Mining for Computer Security. Melbourne, FL, USA: IEEE Computer Society, 2003: 30-39.

[19] ERTOZ L, EILERTSON E, LAZAREVIC A. Detection and summarization of novel network attacks using data mining[C]//Proc of Recent Advances in Intrusion Detection (RAID). Sophia Antipolis, France: Springer-Verlag, 2004: 145-161.

[20] LAZAREVIC A, ERTOZ L, KUMAR V, et al. A comparative study of anomaly detection schemes in network intrusion detection[C]//Proc of the 3th SIAM Conference on Data Mining. San Francisco, CA, USA: IEEE Computer Society, 2003: 108-120.

[21] ERTOZ L, EILERTSON E, LAZAREVIC A. Next generation of data mining[M]. Cambridge, MA, USA: MIT Press, 2004.

[22] ESKIN E, ARNOLD A, PRERAU M. Applications of data mining in computer security[M]. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[23] PORTNOY L, ESKIN E, STOLFO S J. Intrusion detection with unlabeled data using clustering[C]//Proc of ACM CSS Workshop on Data Mining Applied to Security. Philadelphia, USA: ACM Press, 2001: 5-8.

[24] LEUNG K, LECKIE C. Unsupervised anomaly detection in network intrusion detection using clusters[C]//Proc of 28th Australasian Computer Science Conference (ACSC). Newcastle, Australia: ACM Press, 2005: 333-342

[25] OLDMEADOW J, RAVINUTALA S, LECKIE C. Adaptive clustering for network intrusion detection[C]//Proc of the International Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Sydney, Australia: Springer-Verlag, 2004, 3056: 255-259.

[26] WEI F, STOLFO S J, ZHANG J, et al. AdaCost: misclassification cost-sensitive boosting[C]//Proc of the 16th International Conference on Machine Learning (ICML). Bled, Slovenia: Morgan Kaufmann. 1999: 97-105.

[27] WEI F, MILLER M, STOLFO S J, et al. Using artificial anomalies to detect unknown and known network intrusions[C]//Proc of IEEE International Conference on Data Mining (ICDM). San Jose, California, USA: IEEE Computer Society, 2001: 123-130.

[28] LEE W, WEI F, MILLER M, et al. Toward cost-sensitive modeling for intrusion detection and response[J]. Journal

of Computer Security, 2002, 10: 5-22.

[29] BERNASCHI M, GABRIELLI E, MANCINI L V. REMUS: a security-enhanced operating system[J]. ACM Transactions on Information and System Security, 2002, 5(1): 36-61.

[30] FORREST S, HOFMEYR S A, SOMAYAJI A, et al. A sense of self for unix process[C]//Proc of IEEE Symposium on Security and Privacy (ISP). Oakland, CA, USA: IEEE Computer Society, 1996: 120-128.

[31] HOFMEYR S A, FORREST S, SOMAYAJI A. Intrusion detection using sequences of system calls[J]. Journal of Computer Security, 1998, 6: 151-180.

[32] WARRENDER C, FORREST S, PEARLMUTTER B. Detecting intrusions using system calls: alternative data models[C]//Proc of IEEE Symposium on Security and Privacy (ISP). Oakland, CA, USA: IEEE Computer Society, 1998: 133-145.

[33] 饶 鲜, 董春曦, 杨绍全. 基于支持向量机的入侵检测系统[J]. 软件学报, 2003, 14(4): 798-803.
RAO Xian, DONG Chun-xi, YANG Shao-quan. An intrusion detection system based on support vector machine[J]. Journal of Software, 2003, 14(4): 798-803.

[34] QIAO Y, XIN X W, BIN Y, et al. Anomaly intrusion detection method based on HMM[J]. Electronics Letters, 2002, 38(13): 663-664.

[35] CHO S B, HAN S J. Two sophisticated techniques to improve hmm-based intrusion detection systems[C]//Proc of Recent Advances in Intrusion Detection (RAID). Pittsburgh, PA, USA: Springer- Verlag, 2003, 2820: 207-219.

[36] 谭小彬, 王卫平, 奚宏生, 等. 计算机系统入侵检测的隐马尔可夫模型[J]. 计算机研究与发展, 2003, 40(2): 245-250.
TAN Xiao-bin, WANG Wei-ping, XI Hong-sheng, et al. A hidden Markov model used in intrusion detection[J]. Journal of Computer Research and Development, 2003, 40(20): 245-250.

[37] YE N. A Markov chain model of temporal behavior for anomaly detection[C]//Proc of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop. West Point, USA: IEEE Press, 2000: 171-174.

[38] 尹清波, 张汝波, 李雪耀, 等. 基于动态马尔可夫模型的入侵检测技术研究[J]. 电子学报, 2004, 32(11): 1785-1788.
YIN Qing-bo, ZHANG Ru-bo, LI Xue-yao, et al. Research on technology of intrusion detection based on dynamic Markov model[J]. Acta Electronic Sinica, 2004, 32(11): 1785-1788.

[39] 尹清波, 张汝波, 李雪耀, 等. 基于线性预测与马尔可夫模型的入侵检测技术研究[J]. 计算机学报, 2005, 28(5): 900-907.
YIN Qing-bo, ZHANG Ru-bo, LI Xue-yao, et al. Research on technology of intrusion detection based on linear prediction and Markov model[J]. Chinese Journal of Computers, 2005, 28(5): 900-907.

[40] GHOSH A K, SCHWATZBARD A, SHATZ M. Learning program behavior profiles for intrusion detection[C]//Proc of USENIX Workshop on Intrusion Detection and Network Monitoring. Santa Clara, California: USENIX, 1999: 51-62.

[41] MICHAEL C, GHOSH A. Using finite automata to mine execution data for intrusion detection: a preliminary report[C]//Proc of Recent Advances in Intrusion Detection (RAID). Toulouse, France: Springer-Verlag, 2000: 66-79.

[42] MICHAEL C, GHOSH A. Simple state based approaches to program-based anomaly detection[J]. ACM Transactions on Information and System Security (TISSEC), 2002, 5(3): 203-237.

[43] WAGNER D, DEAN D. Intrusion detection via static analysis[C]//Proc of IEEE Symposium on Security and Privacy (ISP). Oakland, CA, USA: IEEE Computer Society, 2001: 156-168.

[44] SEKAR R, BENDRE M, BOLLINENI P, et al. A fast automation-based approach for detecting anomalous program behaviors[C]//Proc of IEEE Symposium on Security and Privacy (ISP). Oakland, CA, USA: IEEE Computer Society, 2001: 144-155.

[45] 童 彬, 秦志光, 贾伟峰, 等. 采用数据挖掘的拒绝服务攻击防御模型[J]. 电子科技大学学报, 2008, 37(4): 586-589.
TONG Bin, QIN Zhi-guang, JIA Wei-feng, et al. A DoS attack defense model adopting data mining[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(4): 586-589.

[46] WESPI A, DACIER M, DEBAR H. An intrusion-detection system based on the Teiresias pattern-discovery algorithm[C]//Proc of EICAR. Aalborg, Denmark: IEEE Computer Society, 1999: 1-15.

[47] WESPI A, DEBAR H, DACIER M, et al. Fixed vs. variable-length patterns for detecting suspicious process behavior[J]. Journal Computer Security, 2000, 8(2/3): 159-181.

[48] WESPI A, DACIER M, DEBAR H. intrusion detection using variable-length audit trail patterns[C]//Proc of Recent Advances in Intrusion Detection (RAID). Toulouse, France: Springer-Verlag, 2000: 110-129.

[49] ESKIN E, LEE W, STOLFO S J. Modeling system calls for intrusion detection with dynamic window sizes[C]//Proc of the DARPA Information Survivability Conference and Exposition II (DISCEX II). Anaheim, CA, USA: IEEE Press, 2001: 165-175.

[50] FENG H P. Dynamic monitoring and static analysis: new approaches for intrusion detection[D]. [S.l.]: Massachusetts Amherst University, 2005.

[51] SOMAYAJI A, FORREST S. Automated response using system-call delays[C]//Proc of the 9th USENIX Security Symposium. Denver, Colorado, USA: USENIX 2000: 185-197.

[52] LIAO Y, VEMURI V R. Using text categorization techniques for intrusion detection[C]//Proc of USENIX Security Symposium. San Francisco, California, USA: USENIX, 2002: 51-59.

[53] KANG D K, FULLER D, HONAVAR V. Learning classifiers for misuse detection using a bag of system calls representation[C]//Proc of IEEE International Conference on Intelligence and Security Informatics (ISI). Atlanta, GA, USA: IEEE Computer Society, 2005: 511-516.

[54] KANG D K, FULLER D, HONAVAR V. Learning classifiers for misuse and anomaly detection using a bag of system calls representation[C]//Proc of the 6th IEEE

Systems, Man, and Cybernetics Workshop (IAW). West Point, NY, USA: IEEE Press, 2005: 210-225.

[55] SUN H M, LIN Y H, WU M F. API monitoring system for defeating worms and exploits in ms-windows system[C]// Proc. of 11th Australasian Conference on Information Security and Privacy (ACISP). Melbourne, Australia: Springer-Verlag, 2006, 4058: 159-170.

[56] 冯 力, 孙 杰, 周晓明, 等. 基于Windows Native API 序列的异常检测模型[J]. 西安交通大学学报, 2006, 40(4): 406-410.
FENG Li, SUN Jie, ZHOU Xiao-ming, et al. Anomaly detection model based on windows native api sequences[J]. Journal of Xi'an Jiaotong University, 2006, 40(4): 406-410.

[57] WAGNER D, SOTL P. Mimicry attacks on host-based intrusion detection systems[C]//Proc of ACM Conference on Computer and Communications Security (CCS). Washington, DC, USA: ACM Press, 2002: 255-264.

[58] KRUEGEL C, KIRDA E. Automating mimicry attacks using static binary analysis[C]//Proc of 14th USENIX Security Symposium. Baltimore, MD, USA: USENIX, 2005, 14: 11-16.

[59] CHEN S, XU J, SEZER E C, et al. Non-control-data attacks are realistic threats[C]//Proc of 14th USENIX Security Symposium. Baltimore, MD, USA: USENIX, 2005: 177-192

[60] 武 斌, 郑康锋, 杨义先. Honeynet中的告警日志分析[J]. 北京邮电大学学报, 2008, 31(6): 63-66.
WU Bin, ZHENG Kang-feng, YANG Yi-xian. Analysis of alert correlation in honeynet[J]. Journal of Beijing University of Posts and Telecommunications, 2008, 31(6): 63-66.

[61] BISHOP M, DILGER M. Checking for race conditions in file access[J]. Computing Systems. 1996, 9(2): 131-152.

[62] KRUEGEL C, MUTZ D, VALEUR F, et al. On the detection of anomalous system call arguments[C]//Proc of 8th European Symposium on Research in Computer Security (ESORICS). Gjovik, Norway: Springer-Verlag, 2003: 236-343.

[63] KRUEGEL C, MUTZ D, ROBERTSON W, et al. Bayesian EVENT CLASSIFICATION FOR INTRUSION DETECTION[C]//Proc of Annual Computer Security Applications Conference (ACSAC). Las Vegas, Nevada, USA: IEEE Computer Society, 2003: 14-14.

[64] MUTZ D, VALEUR F, VIGNA G, et al. Anomalous system call detection[J]. ACM Transaction on Information and System Security (TISSEC), 2006, 9(1): 61-93.

[65] KRUEGEL C, VIGNA G. Anomaly detection of web-based attacks[C]//Proc of ACM Conference on Computer and Communications Security (CCS). Washington, DC, USA: ACM Press, 2003: 251-261.

[66] ROBERTSON W, VIGNA G, KRUEGEL C, et al. Using generalization and characterization techniques in the anomaly-based detection of web attacks[C]//Proc of the 11th Annual Network and Distributed System Security Symposium (NDSS). San Diego, CA, USA: ACM, 2006: 251-260.

[67] TANDON G, CHAN P K. Learning rules from system call arguments and sequences for anomaly detection[C]//Proc of ICDM Workshop on Data Mining for Computer Security (DMSEC). Melbourne, FL, USA: IEEE Computer Society, 2003: 20-29.

[68] TANDON G, CHAN P K. Learning useful system call attributes for anomaly detection[C]//Proc of the 18th International FLAIRS Conference. Clearwater Beach, FL, USA: AAAI Press, 2005: 405-411.

[69] TANDON G, CHAN P K, MITRA D. Data cleaning and enriched representations for anomaly detection in system calls[C]//Proc of Machine Learning and Data Mining for Computer Security: Methods and Applications. London, UK: Springer-Verlag, 2006: 137-156.

[70] 黄金钟, 朱淼良, 郭 晔. 基于文法的异常检测[J]. 浙江大学学报(工学版), 2006, 40(2): 243-248.
HUANG Jin-zhong, ZHU Miao-liang, Guo Ye. Anomaly detection based on grammar[J]. Journal of Zhejiang University (Engineering Science), 2006, 40(2): 243-248.

[71] BHATKAR S, CHATURVEDI A, SEKAR R. Dataflow anomaly detection[C]//Proc of IEEE Symposium on Security and Privacy (ISP). Berkeley, California, USA: IEEE Computer Society, 2006: 48-62.

编 辑 熊思亮

杨宏宇，教授。2003年在天津大学获计算机应用技术专业博士学位，2004～2005年作为高级访问学者赴瑞士洛桑联邦理工大学和苏黎世联邦理工大学从事信息安全研究。现为中国民航大学计算机科学与技术学院教授、中国民用航空局网络与信息安全专家组专家。主要从事网络与信息安全、民航信息系统理论分析与设计方面的研究，在国内外学术刊物发表论文40篇，编写学术专著1部(本)，参与或主持完成国家863项目1项、国家自然基金项目1项、省部级科技基金项目16项，目前主持国家自然科学基金项目1项、国家863重点项目子课题1项、天津市科技支撑重点项目1项、民航科技基金课题4项。曾获省部级科技成果二等奖2项、省部级科技成果三等奖3项。