密级：_____

中国科学院大学
University of Chinese Academy of Sciences

博士学位论文

意图驱动的内部威胁检测技术研究

作者姓名：_____ _____

指导教师：_____ _____

_____中国_____ _____研究所_____

学位类别：_____工学博士_____

学科专业：_____网络信息安全_____

研 究 所：_____中国科学院计算技术研究所_____

2014年6月

Research on Insider Threat Detection Technology for Intention Understanding

By

Chen Xiaojun

A Dissertation Submitted to

University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the degree of

Doctor of Philosophy

Institute of Computing Technology

Chinese Academy of Sciences

June,2014

声　明

我声明本论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，本论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

作者签名：陈小平　　日期：

论文版权使用授权书

本人授权中国科学院计算技术研究所可以保留并向国家有关部门或机构送交本论文的复印件和电子文档，允许本论文被查阅和借阅，可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本论文。

（保密论文在解密后适用本授权书。）

作者签名：陈小平　　日期：

导师签名：　　日期：

摘　要

随着网络信息系统的普及，网络安全变得越来越重要。近年来，一类重要的安全威胁，来自内部人员的攻击给企业、组织和政府机构带来了极大的危害。内部威胁不仅带来重大的经济损失，而且常常引起巨大的负面影响。比如2010年维基解密(Wiki Leakage) 曝光的阿富汗日志事件，2011年中国的CPI曝光事件，2013年的斯诺登事件等等都属于著名的内部威胁例子。

内部威胁的检测防范非常困难。不同于外部攻击，内部攻击者或者拥有信息系统的合法权限，或者能够利用职便窃取他人身份进行非法活动，且其行为具有很强的伪装性和潜伏性。具有合法权限的内部攻击者只能通过检测其行为是否违背预设的安全策略和安全计划来触发告警信息，而高明的内部攻击者往往熟知内部信息系统的安全策略，可以有意识地逃避监控。利用职便窃取他人身份进行非法活动也被称为身份冒用攻击，一般认为，身份冒用攻击者无法完全模拟合法用户的行为特征，可以通过检测细微的生理特征和行为特征的差异来检测身份冒用攻击。

无论是哪种情况，内部威胁的检测需要使用异常检测算法。而在网络安全领域，异常检测算法因较高的假阳性FP(False Positive)饱受诟病。FP会给内部威胁带来昂贵的代价，错误的指控一个内部攻击者不仅浪费资源，并且可能使整个团队的积极性受到影响。更为复杂的是，内部攻击行为表现出很强的潜伏性，分步骤有计划地完成内部攻击的行为很常见。因此，小心翼翼地理解当前正在可能发生的内部攻击，弄清攻击的意图，构建攻击场景，并针对性地加强保护企业的核心资产显得尤为重要。

综上所述，要检测内部威胁，不仅需要解决异常行为检测技术，也要处理异常行为发现过程中引入的不确定性问题，重构多步的攻击场景，理解潜在的攻击意图，并启动保护措施。为此，本文首先研究内部威胁异常行为的感知技术，主要包括身份异常与行为异常的检测技术，然后针对异常检测所引入的不确定性和攻击过程的分步骤性，构建一个概率攻击图模型来刻画内部攻击的过程，在此基础上提出了一套攻击意图推断算法和最大概率攻击路径计算方法，最后扩展概率攻击图使之可以实时计算当前环境下的最优安全防护策略，为系统威胁提供防护建议。

本文的主要创新点如下：

（1）提出了一种基于介入式场景的身份冒用攻击检测方法，该方法通过注入瞬时鼠标失效事件，利用了用户潜意识鼠标行为的特异性来检测身份冒用者。实验表明，该方法在保证较高准确率的同时，可大大缩减鼠标动力学身份认证过程的时间。（2）提出了一个面向内部攻击意图理解的概率攻击图模型，该模型可有效描述内部攻击过程中三类不确定性：攻击者是否有能力发起攻击，攻击行为能否被检测到和攻击步骤能否成功。（3）提出了一种攻击意图推断和场景重构算法，该算法在概率攻击图模型

基础上，根据观测事件序列和相应的置信度，可对目标节点的被攻击概率进行增量计算，并回溯计算最大概率攻击路径。实验证明该算法可有效分析出潜在的攻击目标节点并重构攻击场景。（4）提出了一种最优安全防护策略计算的方法，该方法通过引入安全防护策略节点及其作用、代价属性，将概率攻击图扩展为安全防护概率攻击图，并在安全防护概率攻击图的基础上实现一种最优安全防护策略的计算方法，可在满足一定代价限制的条件下计算最优的安全防护措施集合。

关键词：内部威胁；身份冒用；概率攻击图；意图理解；攻击场景还原；最优防护策略集合

Research on Insider Threat Detection Technology for Intention Understanding

Chen Xiaojun (Network & Information Security)
Directed by Professor Fang Binxing

Due to the increasing usages of information systems over the network, network security becomes more and more important. In the recent years, there was an increase in the number of insider attacks, which caused loss to the private enterprises as well as government. The insider attack not only causes financial loss, but also affects the public images of the enterprises or the government, of which the loss is not easily measurable. There were many insider attacks incidents that attracted public attentions, which include the Wiki Leakage incident reported the Afghanistan events in 2010, the China CPI incident in 2011 and the Snowden incident in 2013.

Insider attacks are difficult to detect and prevent. Different from outsider attacks, inside attackers usually had proper access rights to the information systems, or have obtained the proper identities from other employee in the organization. With proper user identity, the insider can easily hide himself from detection systems. To detect the insider, one possible approach is to analyze the behavior of the user and determine if the user's behavior is violating any security policy or not. But a well-trained inside attacker may have knowledge with the security policy, and he can then perform an attack while not violating the security policy. This attack will make it quite difficult to be identified. If an inside attacker performed an attack using a stolen user identity from another user, it is possible that the attacker is unable to mimic that specific user's personal behavior. It is therefore possible to detect the insider attack by monitor the behavior characteristics of each user. A commonly used technique in insider detection is using "anomaly detection". In network security, anomaly detection usually will produce results with high false positive (FP) rate. High FP rate means a benign employee may falsely be accused as an insider, which may cause high damage to the corporate. Moreover, insider attacks usually performed with a well and structured plan, and being carried out step by step, in a covert manner. It is therefore very important that the attack analysis should be carried out carefully, by identifying the attack target, the intention of the attacker, reconstructing the attack scenario, with the ultimate purpose to protect the valuable digital assets of the corporate.

In order to detect an insider attack, we not only need to detect the anomaly behavior,

but also process the uncertainty caused by the anomaly behavior detector, reconstruct the attack scenario, and then understand the intention of the attacker. In this paper, we shall discuss the anomaly behavior detection in an insider attack, which includes network access anomaly, document access anomaly, and identity impersonation behavior anomaly. Different types of anomalies and different attack paths may introduce different types of uncertainties. In this paper, we present a probabilistic attack graph model to illustrate an insider attack scenario. We then propose an algorithm to estimate the intention of an attack and then determine the attack path the highest probabilities. We then extend the probabilistic model to evaluate the attack graph and determine the best prevention strategy in real-time.

The key contributions of the paper are as follow:

(1) In insider threat detection, we propose in this paper an intrusive technique that is based on mouse behavior. In order to detect potential insider, we propose to perform real-time analysis of an user's mouse behavior when performing user authentication. The scenario-based design and implementation of mechanisms, an insider attack using a stolen ID can be detected with a higher accuracy rate. The user authentication process will take only 5 seconds, which is allowable for practical implementation.

(2) In the understanding of the attack intention, in this paper we propose a probabilistic attack graph model to illustrate the insider attack process. The model is able to show the attack steps and their corresponding causal relationship. Moreover, the model also illustrates the 3 different types of uncertainties: the attacker's capability, the occurrence probability of an attack obtained from observed events, and the success chance of the attack.

(3) In the probabilistic attack graph, we propose two algorithms, one to infer the intention of an attack and another to reconstruct the attack scenario. The two algorithms work together to provide a real-time analysis about the attacker's intention or the probabilities of being attacked for each potential targets. Finally, we can calculate the most likely attack path.

(4) By integrating the protection strategy into the probabilistic attack graph, we can then infer the intention, reconstruct the attack scenario, and construct the real-time hardening measure set in a uniform model. With the real-time security hardening measure algorithm, we can determine the optimal hardening measures to prevent insider attack within the limited available resources.

Keywords: insider threat, identity impersonation, attack graph, intention understanding, attack scenario reconstruction, optimal hardening measures set

目　录

图 目 录

表 目 录

第一章　绪论

“某公司也许购置了能用钱买到的最好的安全技术，员工们也训练有素，每晚回家前把所有的秘密都锁起来，并从业内最好的保安公司雇用了保安，但这家公司仍然易受攻击。一些人可能遵从了专家所有最好的安全建议，安装了各种受推荐的安全产品，并十分谨慎的处理系统配置以及应用安全补丁，但他们仍然很不安全。”By-凯文·米特尼克

1.1　引言

1.1.1　研究背景

随着信息化技术和计算机网络在全球商业、社会、政府、军事以及个人生活中的普及，安全问题便一直是全球化网络的重要威胁。网络安全不仅仅威胁着个人隐私，也威胁到企业的知识产权和国家秘密，给个人和社会带了巨大的经济损失和社会影响。在过去的十来年中，除了普通来自外部黑客的攻击外，来自组织和企业内部人员的攻击威胁也变得越来越严重。在组织或企业内部，恶意的员工为了牟利或者其他目的，窃取知识财产，故意违规操作，进行信息欺诈等行为，使企业形象或者财政严重受损。

1. 2008年，31 岁的法国兴业银行交易员热罗姆.盖维耶尔豪赌股票市场会出现上涨行情，通过伪造账户进了大量的高频小额交易，而在2008年初欧洲市场持续下跌，最终导致其所维护的账户出现巨额亏损，高达71 亿美元，差点导致兴业银行破产。盖维耶尔采用了隐藏方式躲过了系统监控程序，致使管理层发现问题时已经为时已晚[9]。

2. 2010年10月，“维基解密”在一个名为“和平谋杀”的网站上公开了2007 年巴格达空袭时，伊拉克平民遭美国军方杀害的视频。同年7 月，包含7万多份绝密档案的“阿富汗战争日记”被公布。通过长期的调查，美国陆军一等兵曼宁遭到泄密指控。他通过将一张“Lady Gaga”光碟上的原始数据洗去，然后刻录了大量的秘密资料到光盘上，并带出军事基地，随后暴露给了非法的知情者，最后这些资料被“维基解密”曝光。“维基解密”目前还在持续曝光各国政府大量机密的“内部”资料，这些资料包含的范围之广，涉密程度之深在全球引发了一场“地震”。无论是政府、企业对自身数据安全的强烈担忧[10]。

3. 2008年以来，路透社已累计7次精准地“蒙对”了我国的月度CPI数据，总是在中国官方发布之前抢先发布我国的CPI 数据，而且准确度惊人。提前泄露的宏观经

济数据，有可能被一些机构用于提前采取行动规避风险或者谋取利益，严重的还会影响国家经济安全。2011年6月，包括国家统计局办公室一名秘书在内的5名相关人员被指控涉及CPI数据泄露一案[11]。

4. 2013年，前中情局雇员爱德华·斯诺登揭露了一项由美国国家安全局NSA自2007年小布什时期起开始实施的绝密电子监听计划——“棱镜计划”。该计划接管了九家美国互联网公司的数据，以此监视、监听民众电话的通话记录，监视民众的网络活动。从欧洲到拉美，从传统盟友到合作伙伴，从国家元首通话到日常会议记录，其监控范围之广，内容之细令人震惊。棱镜计划惊人规模的海外监听活动引发了一场美国外交“地震”[12]。

上述这些安全事件都具有相同的特点，无论是银行交易员热罗姆.盖维耶尔，美国士兵曼宁，中国国家统计局办公室秘书，还是斯诺登先生，他们都是数据泄密受害方的内部工作人员或者曾经的雇员，他们通过违规操作或者权限不正当使用，将机密信息泄露出去，给企业、组织或者政府造成巨大的财政损失和社会影响。这种威胁被称之为内部威胁(Insider Threat)，相对应的泄密者称为恶意的内部人员(Malicious Insider)。

美国的CSI(Computer Security Institute)和FBI(Federal Bureau of Investigation)每年都对全球的商业公司进行问卷调查，然后从各个方面对计算机犯罪给出一个调查结果。从2000年到2010 年每年的计算机犯罪调查(Computer Crime Survey)统计中，发现内部威胁在各个企业中发生的频率很高，而且造成的损失和危害也特别大。

表 1.1: 2000年至2007年CSI/FBI关于内部威胁引起的经济损失的调查

Year	System Penetration($)	Insider Abuse($)	Unauthorized Access($)
2000	7,040,000	27,984,740	22,554,600
2001	19,066,600	35,001,650	6,064,000
2002	13,055,000	50,099,000	4,503,000
2003	2,754,400	11,767,200	406,300
2004	901,500	10,601,055	4,278,205
2005	841,400	6,856,450	31,233,100
2006	758,000	1,849,810	10,617,000
2007	6,875,000	2,889,700	1,042,700

从表1.1 和1.2 中可以看到，在信息系统面临的各类安全问题中，内部攻击不仅每年都会造成数百万美元的损失，而且发生的频率相当之高。比如2007年，参与调查的企业中，高达59% 的被调查者报告存在内部人员乱用资源的情况，并造成两百多万美元的损失。

表 1.2: 2004 年至2010年CSI/FBI关于内部威胁发生频率的调查

Year	System Penetration	Insider Abuse	Unauthorized Access [1]
2004	17%	59%	37%
2005	14%	48%	32%
2006	15%	42%	32%
2007	13%	59%	25%
2008	13%	44%	29%
2009	14%	30%	33%
2010–2011	11%	25%	29%

1.1.2 研究目标与意义

来自内部的威胁形势日趋严重，内部威胁的防护和检测手段非常重要。首先，内部威胁的检测是系统安全的保障，内部威胁检测对于及早发现内部攻击行为，防范可能的攻击，减小信息泄露风险具有重要意义。其次，内部攻击过程中的意图推断有助于网络安全管理员根据攻击告警和异常告警掌握内部威胁的风险状态，推断核心资产受到攻击的可能性，预测可能的攻击事件。最后，内部威胁的攻击过程场景重构能够提高事后的攻击取证能力，减少误判带来的各种损失。因此，检测内部人员异常行为，推断可能的攻击意图，重构攻击过程和攻击场景，对于保护信息系统，合理规划防护策略具有重要意义。

1.2 国内外研究现状概述

1.2.1 内部威胁的定义

内部威胁伴随着信息系统的广泛应用而一直长期存在。早在20世纪70年代，美国计算机安全专家就指出要重视对系统内部授权用户的约束和防范。随着互联网络及各种应用的广泛普及应用，这种由内部用户导致的安全威胁更加严重。从1999年以来，有很多研究组织和大学关注对内部威胁的研究，其首要的问题是弄清楚那些人员属于内部人员以及什么是内部威胁，研究界对其一直缺乏一致的定义。各个研究者基于自己的数据集和假设，提出了不同的内部威胁的定义。近年来，对内部威胁的定义主要出现了以下几种。

1. RAND的研究人员定义内部人员为“有能力违反公司安全策略或者规则的实体”以及“可以访问敏感信息和信息系统的可信任的人”。直观地说他们将内部攻击者定义为“安全边界内的任何人”，这种定义方法忽略了不同内部人员的可信性与对组织的了解[4] 。

2. Bishop 的一系列文章[13–15] 对内部威胁的定义有比较明确的描述。他认为“内部攻击者指那些可信的并有能力违背给定的安全策略或规则的人，而当这种权利被滥用时，内部威胁便发生了。安全策略特指公司所使用的访问控制策略”。从这个角度出发，Bishop进一步将内部威胁的动作分为两类：合法权限下的安全策略违背行为，比如将手中的资料泄露给第三方；通过获得未授权的访问权限进行非法活动，比如通过盗取别人的帐号密码或者其他手段获取root 权限进行越权操作。

3. 美国国防部高级研究计划局DARPA(Defense Advanced Research Projects Agency)在2007年的项目CINDER 中对内部威胁的定义为：环境内部任何试图完成恶意任务或者达到破坏目的的任何行为都是内部威胁。其中环境可以指计算机系统、网络、通信媒介和基础设施等[16]。

4. Pleed 和其学生Pfleeger的文章中明确将内部人员定义为“具有合法权利访问组织的计算机或者网络的用户”，而内部威胁定义为“一个内部人员的动作，该动作会将组织的数据，资源以一种破坏性的或者不受欢迎的方式置于某种风险之中”[17][18] 。

5. Cappeli等人在2009 年的定义最为准确，在本文中默认他们的定义，他们认为：

   定义1.1. 恶意的内部人员*(Malicious Insider)*
   是指这样的前或者现职员、承包商或商业伙伴：1)被授权访问组织的网络、系统和数据；2)故意超越或滥用访问权限；3)行为损害该组织的信息或者信息系统的机密性、完整性和可用性。
   *来自恶意内部人员的攻击威胁被称为**内部威胁(Insider Threat)**[19]。*

1.2.2 内部威胁的分类

1.2.2.1 攻击意图分类

从攻击的动机方面来分，内部威胁可以分为以下类：

- 无意识泄密(Misused)：内部人员因为缺乏保密意识，或者因保密措施非常脆弱而造成的泄密行为。比如将敏感的数据经过未加密的通信进行传输，或者在即时聊天工具上讨论一些涉密的信息。这类内部威胁没有明确的攻击意图，但往往在不知不觉的行为中泄露机密数据，对企业或者组织造成伤害。

- IT破坏(IT Sabotage)：指内部攻击者通过越权操作对个人、组织、数据或者系统带来伤害。比如心怀不满的前员工在信息系统中设置逻辑炸弹，破坏企业的关键基础设施，或者影响企业的核心业务系统可用性，造成严重的经济损失和社会影响。

- 内部欺诈(Fraud)：指内部攻击者为了盈利而操作或者修改信息系统的关键数据。比如考试系统中老师利用权限帮学生修改成绩单，或者银行系统内部人员利用职权擅自修改帐单信息。

- 间谍(Espionage)：间谍行为在激烈的商业竞争和政府体现比较多，多指由特定组织有计划地派间谍人员打入敌方内部窃取信息情报的行为。

- 知识产权窃取(IP Theft)：指恶意的内部人员为了利益偷取雇主的知识产权。比如有能力的员工因为对待遇或者职位不满将自己在职期间为企业开发的程序源代码据为己有，开设自己的公司，与原企业竞争。

1.2.2.2 内部威胁的分类学

分类学(Taxonomy)是认识复杂事物的一个强有力的思维方式，也是一门综合性学科。一个复杂的事物通常可以分解为多个独立的部分或者属性分别认识，然后再形成一个统一的认识。内部威胁的分类学是为了更好地理解内部威胁，从攻击发生的原因、过程、结果和意图等多个方面认识内部威胁。

内部威胁的分类方法有很多，其中主要的有以下：

1980年Anderson 在研究入侵检测问题时，将计算机系统面临的威胁分为外部渗透、内部威胁和不法行为三种。其中对内部威胁，根据用户访问范围的不同对用户分为了三类："Masquerader"，指利用信息系统的授权漏洞获取合法用户权限的人；"Misfeasor"指乱用职权者，滥用信息系统的功能和数据信息；"Clandestine"指与授权用户相勾结，利用特权绕过审计、控制和访问资源机制的用户。这种分类法仅考虑了最基本的访问范围，没有考虑用户的其他属性，不够完善[20]。

Neumann等人则根据用户的访问类型，将内部用户分为逻辑用户(Logical)和物理用户(Physical)，逻辑用户指那些在安全边界网络外能访问系统资源的人，包括外包人员，合作方等等，物理用户指属于物理网络的区域划分范围以内的人员。这种划分方法中，物理用户的概念无法处理当前越来越普遍的移动网络环境，失去定义原来的意义[21]。

Tuglular等人给出一个基于三维特征的分类框架，首先从三个维度来刻画内部威胁：事件，响应和后果。面临具体的案例时，可以对三维特征进一步细分。例如，事件特征可以进一步分为事件涉及的攻击目标、事件的主体、事件发生的方法、事件发生的地点和时间等，其中的攻击目标还可以进一步划分。这种分类方法不仅可以对用户的行为等进行有伸缩性的深度分类，而且可以对每类的信息粒度进行控制。同时，这种分类办法也对研发内部威胁监控工具有一定的指导作用。不过Tuglular的分类是以响应为中心的，忽视了其他方面的分析，不能较好地对内部威胁进行评估和预测服务[22]。

在Tuglular的分类框架基础上，Magklaras等人[1]以人为中心对内部威胁进行分类，认为所有的动作以及误用操作都能归结到人的因素，具体的分类组织如图1.1所示。这种分类方法也考虑了三个方面的因素。首先，用户角色，指用户在计算机系统中所处的系统角色，主要根据用户对系统知识的掌握程度，可分为三个级别，包括对系统资源有完全访问权利的超级用户，没有管理员权限但掌握大量系统知识的高级用户，以及普通权限和资源的普通应用用户。其次，权限滥用的理由，可分为蓄意和偶然的。蓄意的权限滥用可能包括机密数据窃取，个人差别等。而偶然的事件原因包括对系统、规则的不了解或者由于生活和工作的压力下出错。最后，攻击结果，在发生了内部攻击后，攻击者可能会在系统的各种层面留下痕迹，包括OS 系统的异常日志，网络使用的变化，或者硬件的配置改变等。

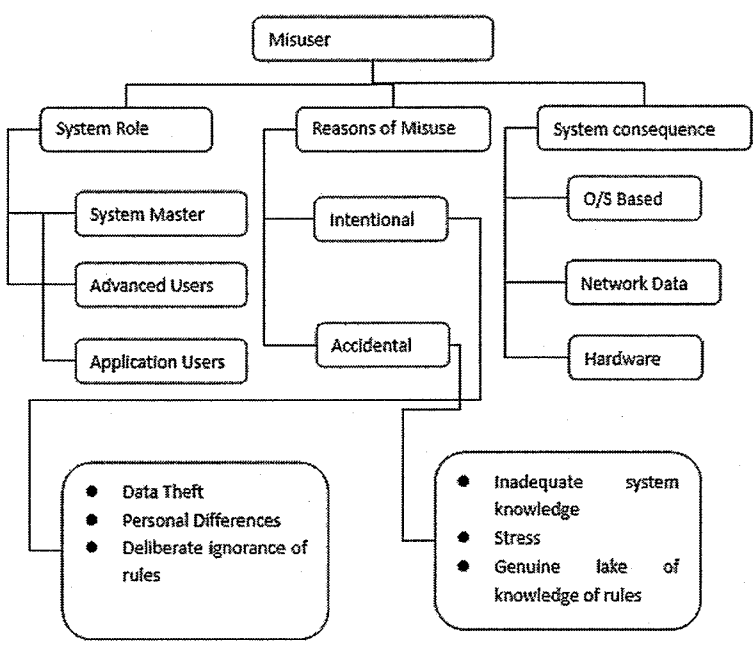


图 1.1: Magklaras等人的内部威胁分类模型[1]

Phyo等人认为上述的分类法都不太适合直接用来构造检测工具，为此，他们按照系统能监控的事件层级将内部威胁分为四大类：a)网络层事件，包括下载超过知悉范围的资料、聊天工具泄密等等；b)操作系统层事件，包括存储敏感数据，输出重定向，未授权软件安装等；c)应用层事件，应用层指严格监视指定列表内的程序运行行为；d)数据层包括文件内容访问检测和数据库访问检测等。根据Phyo等人的分类可以有针对性的构造不同层次的内部威胁检测工具[23]。

近年来的方法Hansman[24]，Kandias[25]也与上述方法基本类似，在此不再赘述。

1.2.3 内部威胁的检测

内部威胁检测的研究从90 年代以来经历了几个阶段，早期的研究关注于内部威胁者的动机、组织环境和实施能力，从个人或者组织的社会属性角度研究内部威胁的动机和如何制定策略减少内部威胁；后来的研究者借鉴了外部攻击中异常检测的方法，开始从不同的操作层面对用户的个人使用习惯进行建模，以尽早发现用户的异常行为或者是身份异常的行为；而最近的研究开始从不同的角度对内部威胁进行全方面的理解和研究。大致来说，内部威胁的检测方法可以分为以下几种：

（1）人、组织与系统动力学

一种对于内部威胁的检测方法采用从个人的动机，能力，机会，资源，权力等角度对当前组织可能的内部威胁进行建模。Schultz用CMO模型来理解内部威胁[26]，即威胁的发起者必须有能力(Capability)，动机(Motivation)，机会(Opportunity)来发起一个攻击。Robinson从五个角度对内部威胁进行建模，分别为：攻击者，意图，目标，动作以及结果。美国国家互联网应急响应小组CERT(Community Emergency Response Teams)总结了不同的行业，比如银行，政府，企业，发生内部威胁时个人的性格以及对企业的危害方面。上面的这些分类都较多地考虑在内部威胁中人为的因素。人为的因素在内部威胁中占据如此重要的位置，Moore等人从系统动力学(System Dynamics)的角度对内部威胁的人进行了建模。系统动力学是从系统理论(System Theory)、控制论(Cybernetics)、伺服机械学(Servo-mechanism)、信息论(Information Theory)、决策理论(Decision Theory)以及电脑模拟(Computer Simulation)等理论所发展出来的。系统动力学中最具特色的是系统中不同的实体（人或者状态）能通过各种信息流互相反馈交流，或者加强或者减弱。Moore 等人用这个概念来模拟现实中内部攻击者发起攻击的原因、影响因素以及各种因素之间的关系。他们在文献[27] 中通过系统动力学描述了知识产权窃取攻击中的两种模型，称为Entitled Independent Model和Ambitious Leader Model，分别用于刻画普通员工跳槽或者自开公司为目的的机密信息窃密行为和Leader偷取公司机密信息的行为，该Leader也许被外部收买，成为竞争对手的一个代理。

（2）基于系统弱点的攻击图模型

与从人、组织及系统动力学的角度进行建模不同，另外一些研究者对攻击的目标或者信息系统的漏洞进行分析建模，将用户可能的行为看作攻击图或者操作树，树的根节点一般指向攻击目标或者需要监控的行为。Upadhyaya 等人将用户当前的操作意图转化为一张SPRINT（Signature Powered Revised Intrusive Table）$< \hat{r}, m >$，然后根据表项建立用户的操作树，而将严重偏离该操作树的动作视为异常操作[28] 。Ray等人将漏洞检测中攻击树技术和SPRINT技术结合起来，形成一棵被裁剪的攻击树，然后通过给每一个节点分配一个概率属性，提出一种概率模型来预测当前用户操作可能的攻击概率[29] 。Butts等人在攻击图模型的基础上，提出了“保护状态”，包括了在

组织的政策和访问控制策略允许范围内的所有活动。任何一次权限的变化都会改变状态，而该模型的目标是为了捕获所有未授权的的变化[30]。Chinchani 等人提出一种全新的KG (Key Challenge Graph)图模型，该模型将信息系统一切可以访问的资源定义为关键(Keys)集合，用节点表示关键，用节点之间的有向边表示用户的活动，边的起始节点代表用户控制和使用的资源，而边的结束节点代表用户访问的另一资源。通过估计用户活动的总体代价，判断其行为是否异常，从而实现整个信息系统的安全评估检测[31]。

（3）主机的行为异常检测

主机上内部威胁检测主要是根据用户在操作系统上的操作序列建立用户的行为模式，然后识别当前的操作行为是否背离行为模式，从而给出警报信息。常用来建立用户行为模式的数据有在类Unix 和Windows 操作系统两个平台上的系统命令序列分析。Schonlau等人[32]在2001年利用利用Unix的ACCT 审计功能程序收集了一套Unix Shell Commands数据集，包含了70个用户一段时间内（几天或者几个月）的命令序列，每个用户收集大约15,000个命令。

使用这个数据集，Schonlau在文献[33]中提出了四种检测算法，并将算法的实验结果与Davison的增量概率转移模型IPAM(Incremental Probabilistic Action Modeling)[34]和Lane等人[35]提的Sequence-Match算法对比，取得更好的效果。

Maxion 和Tan等人[36, 37]使用朴素贝叶斯分类器，研究了窗口大小与检测结果的关系。Coull 等人[38]用生物信息学中DNA 比对的方法来检测恶意命令。Oka等人[39, 40]认为用户的动态行为不一定只与相毗邻的命令有关，也可能那些相关命令并不是彼此相连的。据此，Oka 基于特征并发矩阵ECM(Eigen Co-occurrence Matrix)提出了层次化网络方法来检测恶意命令块中一段时间间隔内并不相连两个相关事件的因果关系，发现异常情况。Szymanski等人提出一种递归式的方法挖掘命令集中的频繁项，即先对原始数据集挖掘频繁项，对齐编码并重写数据集，然后重复该过程，直到没有新的明显的频繁项出现[41] 。Ye等人也使用其他属性和一些统计量，其包括个别事件的发生，事件发生的频率，持续的时间，以及多个事件的联合分布等。其目标是为了证明是否仅仅事件的频率对于检测异常是充分的理由，或者在给定的时间里，单个事件能否足够充分地指示异常的发生[42]。

（4）蜜罐网络诱捕内部威胁

蜜罐(Honeypot)从上世纪90年代开始成为一个强有力的技术手段用来捕获潜在的威胁与攻击，但是大量的研究都关注在如何确定、捕捉和研究外部威胁，而来自内部的安全威胁则只有少数的文章涉及。Spitzner首先提出了用蜜罐网络(Honeynet)来捕捉内部威胁活动[2]。Honeynet由一系列的蜜罐主机和一个蜜罐网关组成，其结构如下图1.2所示：

所有访问蜜罐主机的流量都必须经过蜜罐网关，因此蜜罐网关上的防火墙或

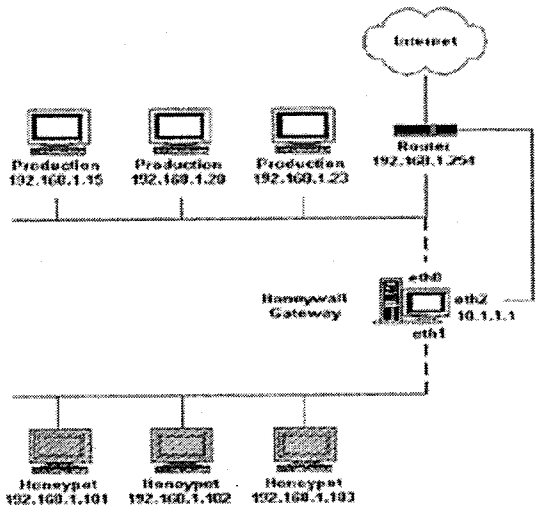


图 1.2: 蜜罐网络系统结构图[2]

者IDS能捕获到所有针对蜜罐主机的活动连接和访问过程。然而为了更好的检测到内部威胁，需要考虑两个重要的问题：a)如何将引导内部攻击者访问蜜罐；b)如何实现与真实环境类似的蜜罐与内部攻击者通信。第二个问题在蜜罐研究中是一个持续重要的问题。针对第一个问题，Spitzner 提出了一个新的概念：蜜饵(Honeytokens)[3]。蜜饵可以是一个文件，也可以是一个信用卡帐号，甚至是任何一段内部攻击者感兴趣的信息，其作用是吸引那些正在扫描和手机敏感信息的内部攻击者。比如针对一个采用sniffer方式偷窥上级管理者邮件内容的内部攻击者者，在传统方式下蜜网技术完全检测不到这种方式的信息窃取，而采用蜜饵技术，可以在内部网络上构造一封类似下图所示的邮件：

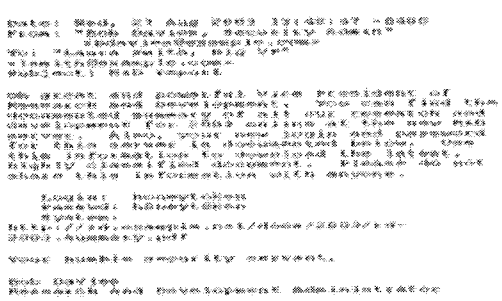


图 1.3: 邮件蜜饵[3]

该邮件蜜饵里的用户名密码，和URL信息都是一个蜜饵，可以引导恶意偷听者访问蜜罐网络上，而后者将详细收集恶意攻击者的信息，了解其攻击意图。

Bowen和Salem等人[43, 44]也针对蜜网技术检测内部威胁做了更深入的研究。他

们的主要研究重点是如何自动生成虚假的诱饵文件(Decoy)，使得诱饵文件的使用能被准确的监测到，而且不会有很高的错误率（即正常用户对诱饵文件的访问），在准确性(TP)和错误率(FN)之间找到平衡。为此，他们设计了一套系统，实现一套诱饵文件的分发系统D3(Decoy Document Distributor)，通过对生成、分发策略的控制，使得诱饵文件保留诱惑性(Enticingness)，难以被发现(Conspicuousness)，可检测性(Detectability)，合法用户的可区分性(Differentiability)，生命周期(Shelf-life decoy)等一系列属性。最后他们实现了在主机和网络对诱饵文件的检测方法，并可以赋予诱饵文件唯一的标识。

1.2.4 内部威胁检测的困难

内部威胁的检测与防护非常重要，然而检测内部威胁却非常困难，这主要体现在以下几个方面[26, 45, 46]：

- 内部攻击者位于系统内部：不像传统的外部入侵，内部攻击者发起攻击的位置位于信息系统内部，直接绕过了传统的防火墙和IDS等传统防护监测系统。
- 内部攻击者具有合法权限：恶意的内部人员具有信息系统的合法方法权限，因此，传统的访问控制策略也会失效。
- 内部攻击行为具有伪装性：位于内部的攻击者还能利用职便，窃取同事或者管理员的身份权限，冒充他人的身份进行恶意的攻击行为。
- 内部攻击行为具有潜伏性：内部攻击者有充分的机会了解内部信息系统，也有充分的时间规划攻击。因次，内部攻击过程呈现多步骤性、潜伏性的特征，要求检测技术能够对攻击过程中的各个步骤进行监控和关联分析。
- 内部威胁错误指控的代价：与外部攻击不一样，针对内部攻击者必须小心翼翼鉴别，在获取真实的证据或者弄清对方的真实意图后才能做出响应，因为错误的指控可能造成研究的团队影响；而对外部威胁，则可采用预防策略，通过禁止某些功能或者服务减少出现事件的可能性。

1.3 本文研究内容及组织结构

1.3.1 研究内容

意图驱动的内部威胁检测技术以对内部攻击过程的检测功能和实时的攻击过程理解、增强实时安全防护能力为目标。它所面临的挑战主要有两个方面：

1. 内部威胁合法权限和合法身份下的攻击行为检测。内部攻击位于信息系统内部，攻击者也可能有合法的访问权限，也极有可能通过职便冒充他人的合法身份信息进行恶意活动。

2. 攻击行为可以通过已知的特征进行检测，但大部分内部威胁检测依靠异常检测算法，无法避免FP的问题，这与内部威胁错误指控的高代价形成冲突。另外内部攻击行为潜伏性也预示着单个的异常行为不能完全揭示攻击者的目标和意图，攻击者可能会辩称其只是由于不小心或者无意识地违背了安全策略。因此，根据单个异常行为指控内部攻击者同样具有很大的错误指控风险。

本文针对以上挑战，研究内部威胁的检测技术，主要目标是：研究内部威胁的检测框架，理解内部威胁检测过程中需要考虑的各种因素及其之间的关系，为接下来的研究提供指导；研究内部威胁单步攻击过程中的身份异常检测和异常行为检测，为尽早发现信息系统内部攻击事件提供良好的数据支撑；研究模拟内部攻击过程的概率攻击图模型，在传统攻击图中引入三类不确定性的讨论和建模，较好地应对内部威胁高FP 和潜伏性所带来的挑战，为完备地表示内部威胁过程提供理论支撑；研究基于概率攻击图模型下的攻击意图理解、攻击场景重构和安全防护策略，理解当前发生内部攻击，攻击的目标，可能攻击的路径，并为网络安全管理员提供安全防护措施建议。

具体而言，本文主要的研究内容包括以下方面：

1. 内部威胁的检测框架研究。结合内部威胁检测问题中所面临的困难和挑战性，研究内部威胁的检测框架，特别考虑影响内部威胁的各种因素，研究内部威胁检测的分层次的意图理解问题，用于指导内部威胁检测技术的有序组织和实施。

2. 内部威胁异常行为的感知技术研究。主要研究内部威胁单步攻击过程中的身份异常检测和异常行为检测。特别针对身份冒用攻击，研究了实用性较好的基于鼠标动力学介入性场景认证方法，在没有降低认证准确性的同时，缩短认证时间。针对短时间窃取数据的内部攻击，研究了文件访问异常模型，检测短期批量下载，周期性扫描等异常行为。该部分的研究用于为后续的意图分析提供数据支撑。

3. 面向内部威胁攻击意图理解的概率攻击图模型研究。为了更好地理解内部攻击者的攻击意图，通过研究攻击图来表述内部威胁的多步骤性，刻画各个攻击步骤之间的因果关系；通过研究攻击图的转移概率，刻画内部攻击发生的三种不确定性，提出概率攻击图模型。然后根据观测事件序列，有效地评估当前攻击者的攻击目标和攻击意图，重构攻击过程，了解攻击发生的场景。

4. 最优安全防护策略研究。基于概率攻击图模型，研究在已发生的攻击事件情况下，在有限的代价限制下，制定安全防护策略，尽可能降低核心资产被攻击的风险，维持系统的可用性。

各研究点之间的关系如图1.4 所示：

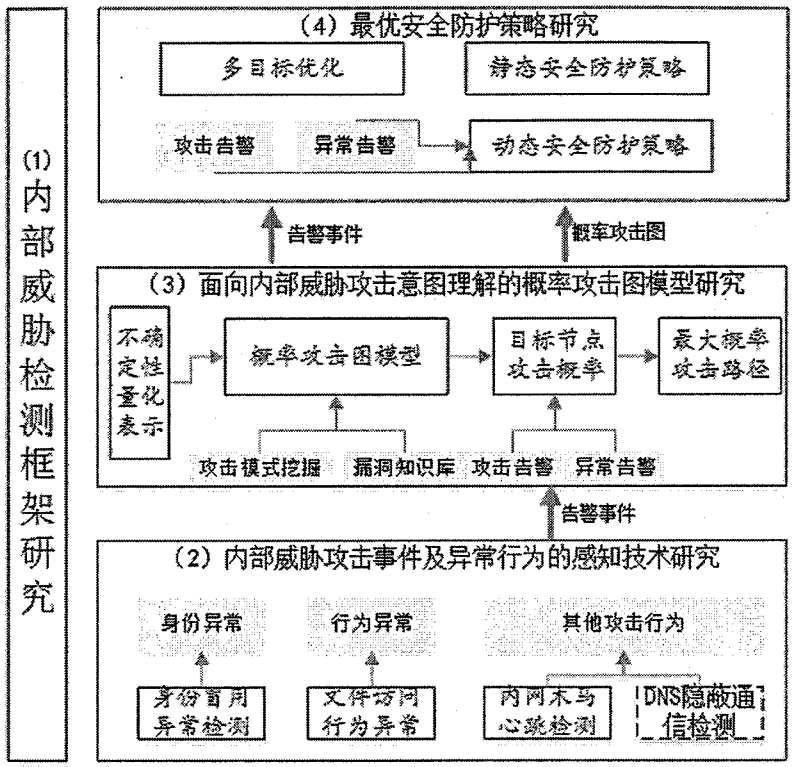


图 1.4: 研究的系统框架

1.3.2 本文组织结构

围绕上述的研究目标和研究内容，本文共分为六章，组织结构如图1.5 所示：

第一章为前言部分，介绍了意图驱动的内部威胁检测技术的研究背景及选题意义，并简要说明了目前对内部威胁的认识、分类、检测方法和困难，说明了本文的研究目标、研究内容及论文的组织结构。

第二章介绍内部威胁检测框架研究，主要介绍了影响内部威胁的各种因素和检测方式，并从异常感知、意图理解和安全防护等三个层次介绍了已有的各种检测方式。这一检测框架被用于指导后续研究工作的有序组织与实施。

第三章介绍内部威胁的异常行为的感知技术研究，主要介绍了身份异常、文件访问行为异常和内部木马心跳行为异常等检测技术。这些异常检测技术针对性地检测不同的异常行为，给出异常告警和相关的参数，为后续的内部攻击意图理解和防护提供数据支撑。

第四章针对内部威胁的多步骤性和异常检测具有的不确定性，研究能够表示内部攻击过程的概率攻击图模型。在利用攻击图表述内部攻击过程之间的因果先后关系的同时，对攻击过程中可能出现的三类不确定性进行详细讨论，并给出取值依据，提出

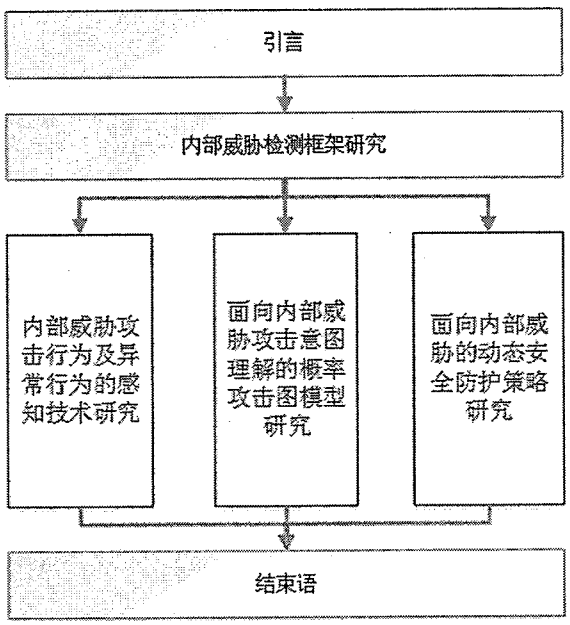


图 1.5: 论文的组织结构

了概率攻击图的形式化描述模型。在概率攻击图模型的基础上，讨论了攻击意图和攻击路径还原等问题。根据已观测事件，评估攻击者潜在的攻击意图或者可能发生下一步攻击，并针对假定的攻击目标，给出了最大概率攻击路径。

第五章在第四章的基础上，进一步讨论攻击发生时的最优安全防护策略计算问题，根据已观测事件和目标节点的攻击概率，在一定代价限制的前提下，计算最优的安全防护策略集，最大限度降低攻击成功的概率，保护核心财产。

最后一章对本文工作及创新点进行了总结，并讨论了该方向下一步工作的重点。

第二章　内部威胁检测框架研究

检测内部威胁需要回答以下问题：

- 内部威胁有哪些影响因素和行为表现？
- 如何检测内部威胁的行为表现？
- 如何通过大量的内部威胁的行为表现来理解内部威胁的攻击意图，并提出防护建议？

本章首先讨论内部威胁的理解，主要从内部攻击的各种影响因素和检测过程进行讨论，然后介绍了当前比较成熟的内部威胁异常行为检测方法，最后为了理解内部攻击意图、计算安全防护策略，介绍了攻击图和日志关联分析方法的研究进展。本章从内部威胁的理解入手，然后形成了异常发现、意图检测和安全防护三个层次的内部威胁检测框架，用于指导后续研究工作的组织和开展。

2.1　内部威胁理解

2.1.1　内部威胁影响因素分析

“一次内部攻击跟哪些因素有关？应该从哪些方面来认识内部威胁？攻击过程会产生什么样的行为表现？”这是本节内容想要解答的问题。

对内部威胁的理解主要与内部威胁的分类学有关，在第一章的内部威胁中介绍了很多内部威胁的分类。本节的内容在前人的工作基础上，从攻击者的视角，从主体、客体和行为三个角度来理解内部威胁。影响因素主要分为三类：攻击的主体即攻击者，用**攻击者的角色**来表现与攻击者相关的各种因素；攻击的客体即被攻击的目标，用**攻击意图**来表示攻击的目标；攻击者的行为表现可由安全监控软件的**观测事件**来刻画。具体的描述如下：

- 攻击者的角色(Role)。用户的角色决定着内部攻击者所拥有的能力和权限。角色又分为两类，一种是组织角色(Organizational Role)，比如管理员，财务人员，数据分析师或者软件工程师，公司的重要管理人员。组织角色基本上从企业或者组织的行政上决定该用户能够接触到的信息范围，对信息系统的使用权限，所拥有的职务便利条件等。另一种角色称为Cyber角色(Cyber Role)，即用户在信息系统中的角色权限，包括比如系统管理员(Administrator)、普通用户(Common User)和客户(Client) 等角色。Cyber权限决定了该用户具体在使用网络信息系统时，所

拥有的实际访问权限。虽然组织角色也决定了用户能够接触的信息范围以及访问信息系统的权限，但行政上规定的访问权限和实际信息系统的访问权限往往是不一致的，对运作不规范的大量小企业来说更是如此。

- 攻击意图(Intent)。攻击意图可以理解为由一系列为了达到某种目的而进行的攻击操作所构成的过程。在内部威胁场景中，攻击可能由蓄意的恶意企图驱动，也可能由正常的非故意行为导致。恶意企图包括：a)机密文档窃取，比如偷取敏感文件、机密数据，或者从SVN中获取重要的源代码数据；b)个人隐私数据窃取，比如偷看个人信息系统的帐号，个人邮件或者聊天记录等；c)安装木马后门为后续攻击准备，这种情况在APT攻击中尤为常见；d) 欺诈；e)IT破坏。正常的意图包括文档信息处理、社交网络、娱乐活动或者系统维护等。

- 观测事件(Observables)。观测事件是指伴随着攻击过程的发生，人们能观测到的各类事件。观测事件又能从主机、网络、应用和其他事件的角度进行划分。主机事件包括系统日志，键盘记录数据，进程序列和I/O访问记录等；网络事件包括端口扫描，邮件通信，Web访问和隐蔽通道等；应用层的事件包括应用系统的日志，比如SSH攻击、数据库访问记录，Web访问记录等。除了从这些角度能观测到各种事件外，还能从其他的IDS系统、蜜网系统中获取到各种观测事件。

内部威胁理解框架的完整组织关系如图2.1所示。

针对观测事件，Anderson等人[4]在2004年对攻击行为的表现进行了详细的分类。他们将IC(Intelligence Community)系统分析师可供观察到的动作分类，包括a)侦察，指攻击前的信息扫描，如Web浏览或者数据库搜索；b)强化，比如安装未授权软件进行后续监控；c)漏洞攻击，利用系统存在的漏洞进行攻击；d)敏感信息的提取与泄露，比如打印、下载或者拷贝到移动存储设备中；e)通信，比如利用加密邮件、隐蔽通道传输；以及f)对抗措施，比如磁盘擦除等操作。如图2.2所示。

2.1.2 内部威胁的检测过程

内部威胁的检测算法就是在各种可供观察到的事件上建立监控点，实时监控各种日志，以发现内部攻击事件。Anderson 等人进一步讨论了内部威胁检测问题，他们认为相关研究问题都落在以下六个方面[4]：

- 用户的角色(Role)
- 活动(Action)
- 观测事件(Observables)

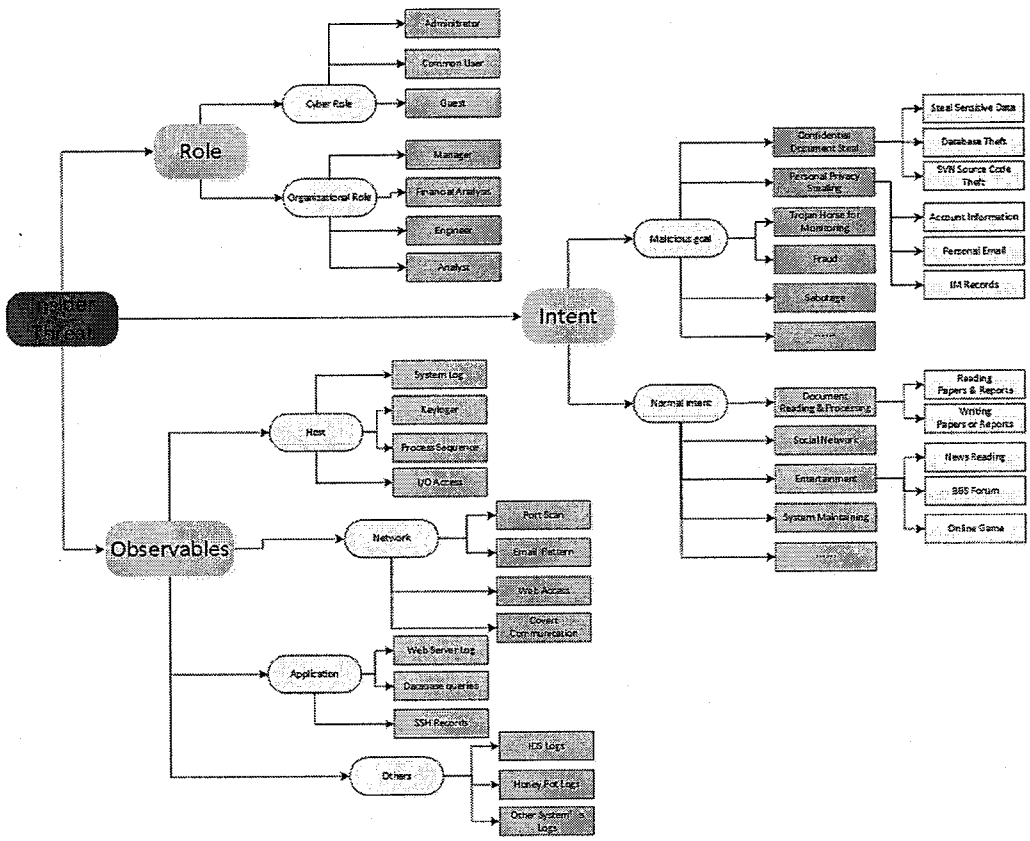


图 2.1: 内部威胁的理解框架

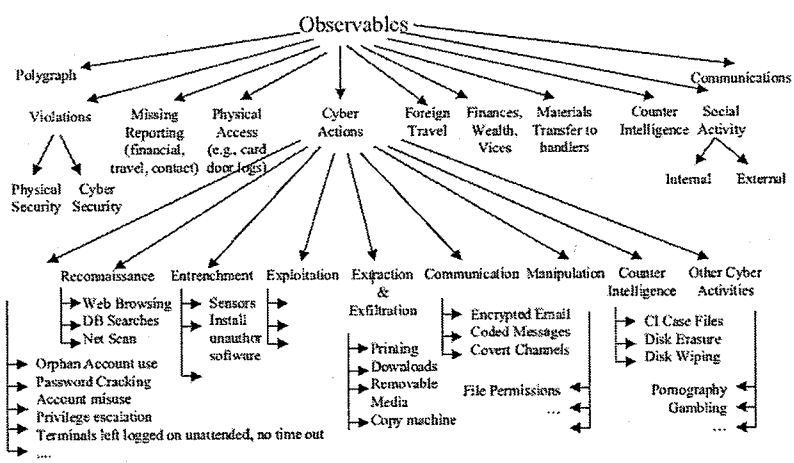


图 2.2: Observables的分类[4]

- 感应器(Sensors)

- 融合和分析(Fusion and Analysis)
- 告警触发行为(Trigger)

系统用户的权限角色决定了用户的活动范围，正常的活动和异常的行为能够被观测到，进而被感应器监控到，监控到的相关事件被送入融合与分析模块进行分析，结果以告警日志的形式告知网络管理员，并进一步采取防护措施。其关键的检测过程如下图2.3 所示：

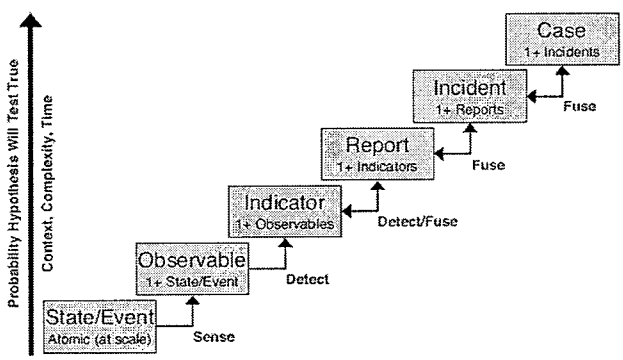


图 2.3: 内部威胁关键的检测过程[4]

在Anderson的研究基础上，本文将内部威胁检测最重要的研究问题集中三个方面：内部威胁异常行为的感知技术（对应于感应器）、内部威胁的意图理解（融合分析）以及内部威胁的安全防护策略研究（告警触发行为）。

- 内部威胁异常行为的感知技术要能够感知到内部网络的各种活动和异常事件，包括主机、系统、网络和其他各种层面的事件，能监控正常的活动、异常活动或者明显的攻击行为。
- 内部威胁的意图理解要求能够将告警事件作关联分析，将多个告警事件与一个攻击场景关联起来，理解用户的目的、意图和攻击过程。
- 内部威胁的安全防护策略要求根据感知技术检测到的各种异常事件，实时制定或者调整安全防护策略，减小攻击风险。

2.2 内部威胁检测中的异常发现

内部网络异常行为的感知技术中最重要的功能是能监控系统中各种观测事件。对于内部威胁而言，具有内部信息系统合法身份的人在攻击时可能产生异常行为，或者利用职便冒充他人身份信息发动攻击。不管何种情况，内部攻击者在其攻击过程中行

为表现可以分为两类：a) 攻击行为，通过雇佣某些攻击工具（比如特种木马等）的行为可能会产生可观测到的攻击行为；b) 行为异常，尽管内部攻击者可能拥有合法的访问权限或者通过伪冒他人的身份来窃取敏感信息，但是可以假定内部攻击者的攻击行为会偏离其正常情况下的行为特征，或者偏离合法用户本身的行为特征，这种情况被称之为行为异常。对内部威胁的异常行为检测，过去已有的工作主要从命令序列行为异常，交互行为异常，文件访问行为异常和数据库访问行为异常等方面来展开。

2.2.1　命令序列异常行为

一个恶意用户或者是窃取他人身份的合法用户在进行攻击时，其执行的操作命令序列会表现出异常行为。命令序列异常主要集中在早期的类Unix的系列操作系统上，很多研究者关注用户命令序列异常分析。最流行的公开数据集是Schonlau等人在2001年左右收集的Unix Shell Commands[32]。这个数据集是作者利用Unix的ACCT审计功能程序收集的，包含了70 个用户一段时间内（几天或者几个月）的命令序列，每个用户收集15,000个命令。为了保护隐私，这些命令都被截断。在数据集中，50个用户被当作入侵目标，其余20个用户被模拟为恶意用户。在50个正常用户中，前5000个命令被认为是正常的，后10000个命令中被随机的注入来自20 个恶意用户的命令块，每个命令块包含100 个命令。当把用户的所有命令分为由100个命令组成的命令块，那么每个命令块要么是正常的，要么是恶意的。问题的目标是准确地识别出恶意的命令块。

该数据集的缺点有：1)从用户收集数据的时间各异，从几天到几个月不等，不同用户登录的次数也都各不一样。2)数据源不清晰，每个人的工作性质不一样，差别很大。3)ACCT审计收集的日志记录是根据进程的结束时间来记录的，因此那些依赖严格序列分析的算法可能完全是错误的。

使用这个数据集，Schonlau在文献[33]中提出了四种检测算法。第一种算法是“Uniqueness”，即用命令在所有用户中的不流行度来识别伪冒攻击者。这种算法基于假设：伪冒攻击者可能使用不常用的命令来实施攻击行为。这种算法只考虑命令的频率，而忽略了时间关系。第二种算法为贝叶斯一步马尔科夫算法(Bayes One-Step Markov)，使用Markov模型引入了两个假设，“NULL 假设”假定当前观察到的命令转移概率可从历史转移概率中计算；“变化假设”假定命令转移变化服从Dirichlet分布。第三种混合多步马尔可夫模型也是在第二种算法基础上扩展开来的。第四种算法是基于压缩的算法，其基本假设是正常用户的命令块应该和训练集有比较相似的模式，将测试集和训练集一起压缩得到的压缩比更高，根据设定的阈值，判断当前测试集是否为恶意命令块。Schonlau将其算法结果与Davison的增量概率转移模型IPAM(Incremental Probabilistic Action Modeling)[34] 和Lane等人提的序列比对(Sequence-Match)算法[35]进行实验对比，取得了更好的效果。

Tan和Maxion等人也研究了窗口大小与检测结果的关系。他们揭示了最好的窗口

取决于测试集中的“外部序列”的最小长度，这是一个未决的先验值。一个“外部序列”指该序列不在训练集中出现，但是序列中的每个元素都在训练集中出现[36, 37]。

Coull用生物信息学中DNA比对的方法来检测恶意命令，他们提出一种半全局对齐的思路，是Smith-Waterman 局部对齐算法的一个修改版。他们开发的记分系统奖励测试数据集中的对齐现象，但是不惩罚未对齐的序列[38]。

Oka认为用户的动态行为不一定只与相毗邻的命令有关，也可能那些相关命令并不是彼此相连的。据此Oka基于特征并发矩阵ECM(Eigen Co-occurrence Matrix)提出了层次化网络方法来检测恶意命令块中一段时间间隔内并不相连的两个相关事件的因果关系，这种关系不能从频率的或者n-grams的角度捕获到[39, 40]。

Szymanski 和Zhang 提出一种递归式挖掘命令集中频繁项的方法，即先对原始数据集挖掘频繁项，对齐编码并重写数据集，然后重复该过程，直到没有新的明显的频繁项出现。他们也用一类SVM来检测恶意用户，使用的特征包括L1和L2 次迭代中频繁项的个数和不同的频繁项个数[41]。

Maxion等人也在另外一个数据集“GreenBerg Dataset“做了试验研究，该数据集在Schonlau数据基础上增加了一些标志和参数。他们利用朴素贝叶斯分类器进行分类，实验证明了方法将TP提升了15%，同时也提高了FP，但是基于Greenberg 数据集实验的ROC曲线整体在Schonlau数据集实验的ROC曲线之上，这表明更多的标志和参数等特征能取得更好的检测效果[47]。

除了以上描述的特征外，Ye等人[42]也使用其他属性和一些统计量。其中包括个别事件的发生，事件发生的频率，持续的时间，以及多个事件的分布等。他们的目标是为了证明是否仅仅事件的频率对于检测异常就足够了，或者在给定的时间里，单个事件是否足够指示异常的出现。他们一共用了五种统计方法，决策树；Hotelling's T2；卡方分布(Chi-Square Test)；多变量测试(Multivariate Test)和马尔科夫链(Markov chain)。实验数据基于Solaris的Basic Security Model收集了250个安全相关的事件。实验结果证明了事件的频率和事件的顺序属性对于检测异常的重要性。

2.2.2 人机交互行为异常

除了人与系统之间的命令序列交互，另外一些人对人机交互行为进行研究，其中最重要的包括人通过鼠标和键盘与系统进行交互。鼠标和键盘的行为特征能够反映人独特的生理行为特征，可被用于异常身份冒用行为的检测。

高艳等人是基于击键动力学来识别身份冒用者，他们统计每个键的延迟时间(键按下去到弹起来之间的时间间隔)和每两个键之间的间隔时间。计算间隔时间和延迟时间的均值和方差。并根据按键的频率赋予权重，频率越高，权重越大，反之越小。将间隔时间和延迟时间的分布看作高斯分布，时时计算出当前统计的概率，如果概率超过特定阀值，则认为是正常用户，否则认为是非法用户。为排除一些时间间隔大的数据，

如用户使用键盘中间与人谈话等，根据用户按键时间的均值计算一个阀值，排除异常数据。他们的实验结果显示误警率为2.61%，误报率为5.73%。

Pusara等人介绍了一种基于鼠标移动数据的用户认证方法，用于检测用户非法使用计算机。该方法的主要思想为：每隔100ms 收集鼠标的位置数据，计算出鼠标的移动速度，方向，距离作为一部分特征，收集鼠标的事件数据，即单击、双击、非客户区移动的数据，同样计算出速度、方向和距离作为特征。设定一个窗口，从每个窗口里计算一个特征代表该窗口内数据的特征，然后采用决策树的方法检测异常值。在实验中，他们记录18个实验者2个小时内使用IE浏览同样的网页点击鼠标的数据作为测试数据，用C5.0作为决策树分类软件。使用平滑过滤器(smoothing filter)后，平均假阳率为0.43%，平均假阴率为1.75%[48]。

Valacich等人从测谎仪的原理中得到启发，发现人的潜意识与行为有一定的关系。特别地，他们发现如果设计一个特定的问答认证测试，要求内部攻击者进行测试(Conceal information test)时，其鼠标行为与正常用户的行为表现得不一致[49]。

2.2.3　文件访问行为异常

基于文件访问行为的内部威胁检测是基于如下的假设：正常用户非常了解自己的文件系统，当其完成工作需要搜索文件时，会以一种有限制的、有目标的方式去搜索文件系统，这种搜索方式具有个人的唯一性特征。与此相反，身份冒用者可能对受害者的文件系统以及桌面布局不是很清楚，从而会以一种比较广的方式来搜索文件，产生昂贵的代价。基于以上假设，Salem 等人设计一套系统，捕获用户对信息的搜索与访问活动，提取了一个很小的仅与搜索操作相关的特征集，然后用一类支持向量机ocSVM(one-class Support Vector Machine)模型给每个用户建立自己的访问模式。他们的试验结果取得了1.1%的假阳率[50]。

Maloof在早前也开发了一套系统来检测侵犯"Need-to-Know"原则的行为。"Need-to-Know"是一个抽象的概念，违背"Need-to-Know"原则指那些恶意内部人员有权访问数据，但是他们从事一些恶意的活动去获取与他们被分配的任务无关信息或者数据。"Need-to-Know"数据可以根据用户的组织信息，通过对用户日常的工作所访问到的数据建模来获得。为此，他们通过监控网络流量，获取用户的信息使用事件，包括文件浏览、搜索、下载或者打印等等。然后结合用户的组织信息和使用文档的上下文信息构建76个的基本检测器，用一个贝叶斯网络模型将所有这些基本检测器的结果联合起来分析，最后他们也给安全分析员提供关于威胁、事件、告警相关的检索接口，协助安全分析员的管理[51]。

Maloof等人和Salem等人的工作都非常具有新意，但也存在很多可改进的地方，比如"Need-to-Know"原则没有处理变化情况，当一个人被分配的任务改变时，其需要访问的数据也会改变，这种任务变化会带来突发的无序情况，可能会被"ELICIT"系统识

别为异常情况。Salem等人的研究假定了攻击者对系统一无所知，但实际情况可能攻击者做了足够的踩点工作，对受害者的文件系统和桌面布局有一定的了解，此时，他们的模型会失效。

2.2.4 数据库访问行为异常

数据库的内部威胁主要来源用户的数据滥用行为，用户可以在合法的权限下访问不需要知晓的信息，或者窃取别的数据库用户帐户信息来获取机密数据。数据库的内部威胁模型主要是对用户的使用模式进行建模，通过比较攻击行为过程中的使用模式来发现异常行为。一般认为，特定用户的使用模型具有相对稳定性，其访问数据行为不会发生太大变化，且攻击者访问行为与正常的访问行为是不一样的。

Kamra 等人用统计的方法对用户的SQL Expression(主要是select queries)进行分析和建模，对不同角色的用户训练分类器[52]。Mathew等人认为利用SQL语法进行建模有一定的缺陷，提出一种以数据为中心的内部威胁检测算法，即对用户每次访问的数据进行建模[53]。该方法的主要特点为：a)是一种以用户具体访问的数据为中心的内部威胁检测方法；b)从检索结果中提取特征的方法只与数据库模式有关，与数据库大小和检索结果大小无关。但是该方法的缺点是不能应付结果聚合的情况，如GroupBy操作。而且只适合静态的数据库表，因为当更新数据时，必须手动修改异常检测的边界。

2.3 内部威胁检测中的意图理解

2.3.1 告警日志的价值困境

由于日益严重的安全问题，大部分企业和组织都已经深刻意识到网络安全防护的重要性，因此，现代网络信息系统往往部署了很多安全防护产品来抵御各种攻击。防火墙入侵检测系统IDS(Intrusion Detection System)从计算机网络系统中的若干关键点收集信息，并分析这些信息，检查网络中是否有违反安全策略的行为和遭到袭击的迹象。入侵防护系统IPS(Intrusion Prevention System)还能阻止可能的攻击过程。杀毒软件，操作系统和网络应用系统都可以设置相应的安全策略，记录可能的安全攻击事件日志。

这些安全产品和防护系统从不同的功能层面监控着系统中各个关键点，根据网络管理员配置的规则或者自身的异常检测能力捕获可能的攻击和报警事件，给出报警信息。但是当前的安全产品大都缺乏对报警信息的后续处理，仅仅是根据自身的检测规则给出最原始的告警信息。原始的告警信息一方面描述的都是最底层语义的攻击检测事件，如图2.4,图2.5，日志量巨大。比如在我们所获得的某组织网络信息系统里，从早上8点到11 点期间内，各种IPS和IDS 中的原始报警数高达为519,790 条，即使对其中的频繁重复出现的同类日志进行了大幅度聚合，仍然有929条高级的告警日志。另外一方

面由于异常行为检测算法本身固有的特性，FP较高，虚假的告警日志较多。这两方面使得网络安全管理员要面临非常大的压力，很难真正理解网络中正在发生什么样的攻击，而对真正的网络攻击做出及时、有效的安全响应就更加困难了。这样，大量杂乱的告警日志处于一种价值不明的状态，缺乏有效手段去分析其价值。这一问题被称为“告警日志的价值困境”。

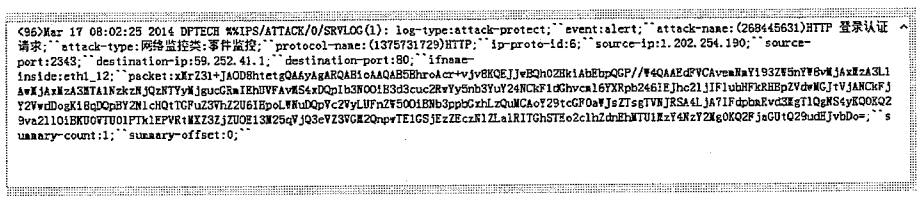


图 2.4: DPTECH产品安全日志

图 2.5: RG-Wall IPS监控日志

为了解决这一困境，网络安全管理员可能希望有自动化的手段对大量的原始安全日志进行分析处理，并回答如下问题：

- What has happened?当前的网络信息系统正在经历何种真正的攻击？
- Who made it? When and how it happened? 攻击是由谁发动的，什么时候发动的，是怎么发生的，下一步攻击目标是什么？即攻击者的攻击意图是什么？
- How to mitigate risk of insider's attack?评估攻击网络信息系统的危害是什么，在当前状态下如何制定防护策略，做到可用性、代价以及风险收益之间的均衡?

第一个问题是为了找到原始告警事件中真正有价值的危险预警信号。针对这个问题，大部分的工作是通过对大量的原始报警事件进行聚合、去除虚假的告警事件和对告警日志确定优先级等方法来解决，称为告警日志预处理技术。第二个问题是即在当前攻击动作发生的情况，理解已经发生的攻击是如何一步步发生的，并预测未来可能的攻击目标和可能采用的攻击路径。将告警日志进行关联分析，重构攻击场景是解决该问题的主要方法。第三个问题是要评估当前的攻击对整个网络信息系统的危害风险程度，并为制定最优化的安全防护策略提供建议，多用攻击图技术对其进行研究。

2.3.2 日志关联分析技术

日志关联分析技术一般要求提供一定的规则对日志进行关联，也被称为基于规则的日志关联技术。主要基于如下基本观察：大部分相关的告警信息，都是因为他们同属于攻击的早期阶段或者是某些高级持续攻击的中间步骤，对于基本的告警日志来说，他们之间有可能存在前提条件关系(preconditions)和后续结果关系(consequences).早期的告警记录可能满足晚期的告警记录的前提条件，晚期的告警记录可能满足晚期告警记录的后续因果关系。

基于该思想，文献[5]将攻击描述为一串抽象的攻击概念的组合体(concepts)，在每一个攻击概念能够实例化之前，必须要求一定的攻击能力(Capabilities)被满足。他们提出了一个模型以及描述该模型的语言JIGSAW，在该模型中，每一个攻击概念必须有一个对应的攻击能力描述规范，该攻击能力描述规范描述了攻击发生必须具备那些攻击能力，以及会产生什么样的后果。图2.6描述了一个RSH_Connection_Spoofing攻击概念的示例。其中Requires表示攻击概念发生前的能力需求，即前提条件，而Provides则表示攻击发生后的后续结果，Action简单表示检测到该规范定义时应该采取的动作。

```
concept RSH_Connection_Spoofing is

    requires
        Trusted_Partner:    TP;
        Service_Active:     SA;
        PreventPacketSend:  PPS;
        extern SeqNumProbe: SNP;
        ForgedPacketSend:   FPS;
    with
        TP.service is RSH,              #- The service in the trust relation is RSH
        PPS.host is TP.trusted,         #- The blocked host is the trusted partner
        FPS.dst.host is TP.truster,     #- The spoofed packets are sent to the trustor
        SNP.dst.host is TP.truster,     #- The probed host is the trustor
        FPS.src is [ND.host,PPS.port] #- claimed source of forged packets is blocked

        SNP.dst is [SA.host,SA.port]    #- The probed host must be running RSH on the
        SA.port is TCP\RSH,             #-   normal port
        SA.service is RSH,              #-

        SNP.dst is FPS.dest             #- probed host must be where forged packets are sent

        active(FPS) during active(PPS)#- forged packets must be sent while DOS is active
    end;

    provides
        push_channel:       PSC;
        remote_execution:   REX;
    with
        PSC.from    <- FPS.true_src;    #- Capability to move code from attacker to RSH server
        PSC.to      <- FPS.dst;         #-
        PSC.using   <- RSH;             #-

        REX.from    <- FPS.true_src;    #- Capability to execute code on remote host
        REX.to      <- FPS.dst;         #-
        REX.using   <- RSH;             #-
    end;

    action
        true -> report ("RSH Connection Spoofing: TP.hostname")
    end;
```

图 2.6: RSH_Connection_Spoofing-RSH连接欺骗例子[5]

Peng Ning等人在文献[6]中认为JIGSAW要求描述每一个攻击概念的前提条件能力理论上可行，但实际上是不可行的。当IDS检测某个报警事件失败时，告警关联将进展不

下去。同时JIGSAW 对所有原始的告警日志来做相似处理，也非常缺乏可行性，因为如果攻击者为了达到某一步攻击做了很多相似的攻击常识，那么JIGSAW只能关联其中的一条原始告警日志。另外，JIGSAW仅仅提供了一个模型和模型的描述语言，并没有提供一套实现机制。Peng Ning等人通过将同一类的原始告警日志归结到一个超级告警类型Hyper Alert Type，即具有相同前提条件和后续结果的原始告警日志集被定义在一个超级告警类型中，一个超级告警类型的实例就是一个原始告警加上一个发生的时间戳。根据所定义的超级告警类型，一个超级告警关联图被构建，如图2.7所示。Peng Ning也实现了一个线下分析工具实现日志的关联并构建攻击场景。

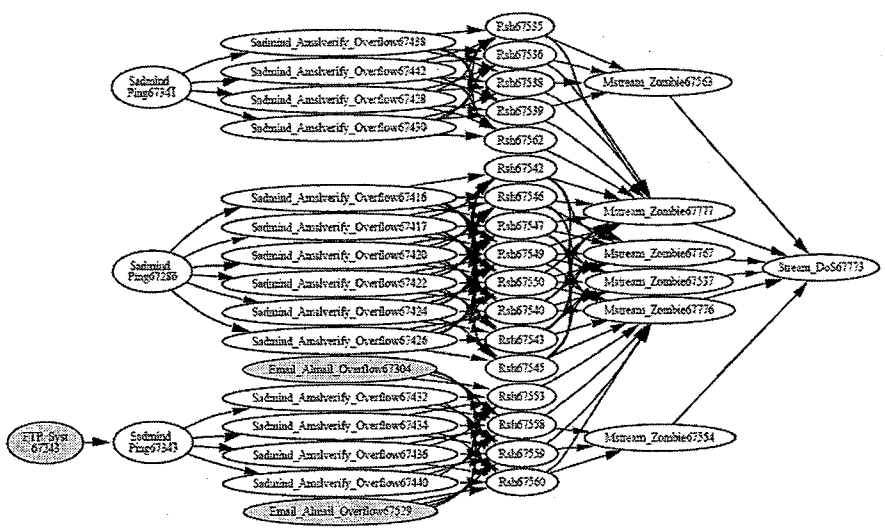


图 2.7: Hyper Alert Correlation Graph 超级告警日志图[6]

虽然基于规则的日志关联方法不要求完全的攻击场景作为先验知识，但也要求特定的攻击知识，比如明确知道某类攻击的前提条件与后续结果。因此，尽管基于规则的日志关联方法能够显式地发现告警日志之间的关联关系，并构建攻击场景，但是这些方法不能发现新的攻击模式，因为新的攻击模式中攻击的前提条件和后续结果是未定义的。

2.3.3　攻击图技术研究

随着网络攻击技术逐步多样化和智能化，攻击者在实施网络非法活动时会针对其存在的脆弱性采取多步骤的组合攻击方式进行逐步渗透，虽然有诸多成熟的脆弱性扫描工具等，能够自动发现目标网络中已知的脆弱性，但是这些工具孤立地研究各个脆弱性，不能分析它们之间的相互作用关系和由此产生的潜在威胁。攻击图技术把研究对象抽象为两个目标主体：目标网络和攻击者，认为目标网络和攻击者之间存在博弈的关系，即目标网络努力保持自己在正常的状态空间转化，而攻击者总是试图使目标

网络向不期望的状态转化。它首先以面向攻击的方式分别对目标网络建模和攻击者建模，然后根据二者之间的相互作用关系产生攻击图，由于该技术能够自动发现未知的系统脆弱性以及脆弱性之间的关系，从攻击的角度展示了攻击者利用网络内存在的脆弱性进行逐步入侵的过程，因此它作为一种新的脆弱性分析技术正成为越来越多研究者关注的焦点之一。

攻击图技术研究大致可以分为攻击图模型的构建、攻击图分析研究以及网络安全措施优化等三个方面的应用研究。

2.3.3.1 攻击图模型构建

Sheyner等人首次实现了一个攻击图模型的自动构建方法[7]。他们定义攻击图为一个四元组$G = (S, \tau, S_0, S_s)$，其中节点集S表示网络的状态，$\tau \subseteq S \times S$，表示状态之间的转移关系，$S_0, S_s$分别表示初始状态节点和攻击节点。他们提出了一套工具来从网络拓扑自动生成攻击图，并进行攻击图上的分析，如图2.8所示。其中，NuSVM是一个模型检测器，用来描述攻击模式之间的依赖关系并生成图模型。Sheyner等人也修改了该模型检测器，并提出用XML规范来代替NuSVM语言来描述网络状况，方便网络管理员的使用。

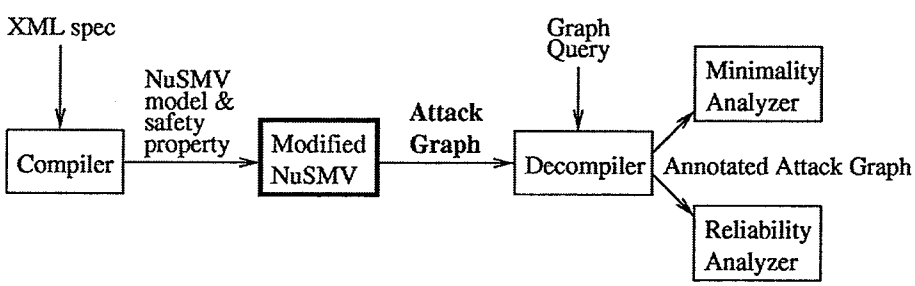


图 2.8: Sheyner等人的攻击图生成与分析工具[7]

为了使得攻击图构造更加自动化，从目标网络中自动获取网络模型参数也非常重要。Jajodia等人利用扫描工具Nessuss[54] 来获取系统中的漏洞，并集成到他们开发的攻击图构建和分析工具TVA中[55]。Ou等人在此基础上继续讨论了防火墙、NAT等问题，他们开发了一套代理程序MulVAL 扫描器来自动从各个主机上采集软件和服务等相关信息[56]。

2.3.3.2 攻击图分析

在攻击图结构与不确定性的研究方面，Lingyu Wang 等人根据漏洞攻击的难度和配置网络的代价，建立了攻击图的概率属性，首次正式地提出了评估网络风险的一种概率安全属性，并在此基础上提出了用累计概率的方法计算整个网络的安全属性。该

方法讨论概率属性的几种计算方法，但并未引入观测事件的置信度进行概率计算[57]。Peng Xie等人第一次深入地探讨了在攻击图中三种不确性的来源，分别为攻击图结构的不确定性，攻击动作发生的不确定性以及触发报警的不确定性。他们在模型中对一些重要攻击节点引入前提条件，即攻击动作节点AAN(Attack Action Node)，计算中只有当攻击动作节点为真时才对攻击节点进行概率推导。[58]。

在攻击图的概率推导计算方面，张少俊等人提出了一个在满足观测事件偏序条件的情况下，利用贝叶斯推理计算攻击图节点的置信度方法。他们的模型中攻击节点与观测事件的条件关系与本文所提的正好相反，该模型根据当前攻击节点的概率，使用似然抽样算法推导观测事件的置信度[59]。在Lingyu Wang等人工作的基础上，叶云等人提出了一种攻击图简化算法来解决存在环时攻击图推导存在的性能瓶颈问题，主要是通过剔除一些攻击图中的不可达路径，减少图中的循环路径，然后给出计算最大可达概率的方法来解决循环路径导致的概率重复计算问题[60]。

2.4　内部威胁的安全防护策略研究

攻击图能够识别目标网路中脆弱性之间的关系以及它们产生的潜在威胁。为了保证目标网络中的关键资产不被攻击者破坏，研究者往往基于攻击图对网络安全措施优化问题进行研究，研究在最小风险、代价限制、系统可用性但多个目标的优化算法，实时制定或者调整安全防护策略。

最优安全防护策略集的研究工作从攻击图被提出便开始了。Jha等人最先在状态攻击图上计算最优安全防护策略，认为每一步原子攻击可以通过安全措施来阻止，他们基于状态攻击图寻找保障目标网络中关键信息资产安全的最小安全措施集。他们将关键集定义为一些原子攻击组成集合：当移除了属于关键集中的原子攻击之后，攻击者不能从初始攻击状态节点到达攻击目标状态节点。而最小关键集即为所有关键集中最小的集合。证明了最小关键集的问题可归结为Hitting Set问题，并提出一种贪心算法来来解决[61]。

Noel等人认为Jha的最大问题是依然有大量无关和冗余的攻击节点被包含到最小关键集中，他们认为阻止原子攻击的最好方式是从源头上把引起这些原子攻击发生的前提条件消除掉，并且每个安全弥补措施需要一定的成本。他们基于属性攻击图提出从源头上把引起这些原子攻击发生的前提条件消除掉。为此他们将攻击图归结为一个CNF合取范式，寻找能防止所有攻击路径发生的最小初始条件集，提出了最小成本的安全弥补措施集。但是该方法不能应用于大型的具有含圈攻击路径的攻击图[62]。

Lingyu等人提出基于逻辑推理的方法，首先把该问题转化为布尔表达式，然后通过求该表达式的析取范式计算出所有的弥补措施集合。并在此基础上求最优弥补集，该方法在最坏情况下具有不可避免的指数时间复杂度，无法应用于大规模目标网络[57]。

吴金宇等人同时研究了最优原子攻击修复集和最优初始条件修复集问题，他们进一步证明了该问题可以归结到最小S-T 割集问题，并提出了一个贪心算法解决该问题。Noel、Wang和吴金宇等人的工作建议从源头消除攻击隐患，这种计算方式能使得关键集最小化。但是通常情况下，要完全消除初始节点的漏洞隐患代价是非常昂贵的[63]。

Poolsappasit 和Dewri等人认为上述安全风险评估都只考虑网络配置的条件，他们将问题域扩展到安全控制手段的代价和收益问题，提出一种多目标优化的分析方法，该方法试图在安全控制代价和整体收益方面寻找一个平衡点。在此基础上进一步提出使用一种基因算法计算多目标优化问题，取得不错的效果。Dewri与Poolsappasit两人的工作介绍了一种贝叶斯攻击图，并给出了一种多目标的安全防护策略计算方法[64, 65]。

2.5 本章小结

本章首先讨论了内部威胁的理解模型，提出从攻击者的角色、攻击者的意图和观测事件三个角度来理解内部威胁，进一步介绍了对攻击过程中可能被观测到的事件的详细分类，明确了内部威胁检测框架中最重要的三个问题，异常发现、意图理解和安全防护。接下来的三节分别介绍了这三方面的工作进展。内部威胁中的异常检测技术研究主要介绍了命令序列异常检测，人机交互异常检测，文件访问行为异常和数据库访问行为异常等检测技术；意图理解介绍了如何根据观测事件日志进行意图理解、场景重构方面的研究，主要包括利用攻击图来检测已知的攻击模式和利用日志关联分析挖掘未知的攻击模式两个方面；安全防护策略研究讨论了在攻击图上如何进行最优安全防护策略的实时计算问题。综上所述，本章主要讨论内部威胁的检测框架，介绍了在框架内其他工作的进展情况，接下来的第三、四、五章都是在该框架展开的具体研究工作。

第三章　内部威胁异常行为的感知技术研究

内部威胁的第一个挑战是内部攻击者在合法权限和合法身份下的发起攻击，通过对身份异常、行为异常、攻击行为的检测技术有助于快速掌握网络信息系统中的异常状况，是发现内部威胁的第一步工作。本章首先讨论了身份冒用问题，介绍了一些可用来检测身份冒用的技术和方法，然后利用鼠标动力学的原理，通过设计介入式场景提出了一套实际可行身份实时认证系统。本章接着讨论内部威胁行为其他异常检测的方法，包括文件访问行为异常和木马心跳通信检测算法，最后总结该章内容。

3.1　基于鼠标动力学的身份异常检测技术研究

3.1.1　身份冒用与身份认证技术

内部威胁已经成为对企业、组织和政府最为重要的威胁来源之一，而且受到学术界、企业和政府相关研究机构的重视。其中身份伪冒者(Masquerader)是内部攻击者中的一种，他们通过人情关系、社会工程学等手段窃取企业组织内重要管理人员或者其他人的信息系统身份账号，或者利用职便使用他人电脑来窃取机密信息。最常见的一个身份伪冒攻击的场景是：信息管理员在下午3点钟出去喝下午茶，然而却忘记了注销其在公司的内部信息系统的登录会话，也没有对Win7操作系统进行锁屏操作。虽然该用户的桌面系统设置了10分钟锁屏选项，但是恶意的同事可能在短短的半个小时下午茶空闲使用其电脑，浏览或者窃取他电脑上的私密文件，或者以信息管理员的权限流量公司内部信息系统的敏感数据，如3.1所示。

作为内部威胁的一种，身份冒用令人防不胜防，现有的主要防御和检测手段就是通过实时身份认证技术对用户进行访问控制和持续性身份认证。具体而言，身份认证技术主要包括基于口令和智能卡的身份认证技术，基于生物生理特征的身份认证技术，基于生物行为特征的身份认证技术，如图3.2 所示。

3.1.1.1　基于口令和智能卡的身份认证技术

基于口令或者智能卡的身份认证方式是最常见的身份认证技术。

- 口令认证。最常见的认证方法，比如操作系统，网络管理系统，以及在互联网上各个网络应用等都采用该种类型的认证方式对用户身份进行认证。该认证方法实现简单，成本最小，在有效的管理下，该方法可以防止恶意攻击者登陆系统，防范信息窃取和信息破坏等攻击事件。但是它的作用范围仅在攻击者无法获取用户口令的前提下，且一般不能对用户进行持续认证。

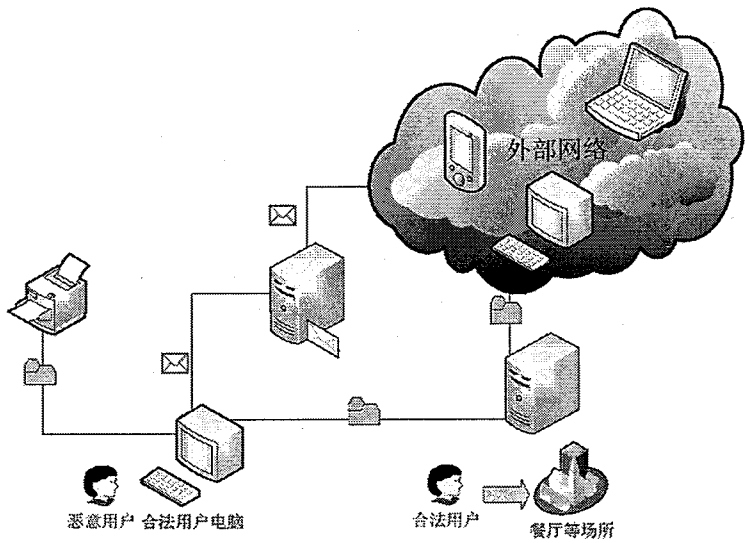


图 3.1: 身份冒用攻击(Masquerader)示例

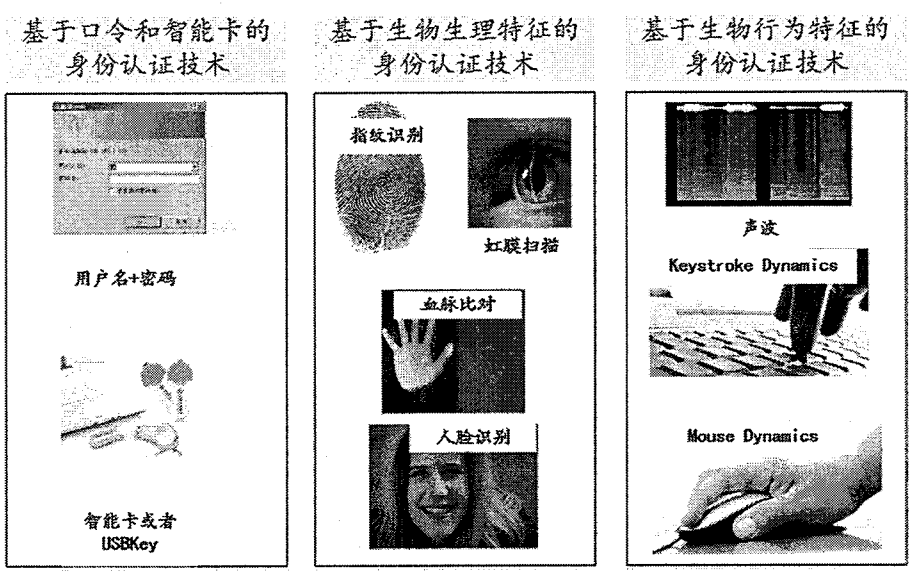


图 3.2: 身份认证技术汇总

- 智能卡认证[66, 67]。智能卡是一种内嵌微芯片的塑制卡片，如各种电子门禁卡，身份卡片等。智能卡通常的原理是通过雇佣一个PKI(Public key infrastructure)服务，卡片里面存储了一个由该PKI颁发的加密数字证书和一些相关信息，读卡器通过读取相关信息进行身份鉴别。USBKey 是另一种形式的实物认证工具，采用一次一密的双因子认证模式，广泛应用于电子政务和网络银行等对安全性要求较高的领域。

基于口令的身份认证和基于智能卡的身份认证都有容易丢失或者被泄露等缺点，一旦遗失或者被泄露就会造成比较大的危害。特别是口令安全最近已经成为互联网面临的最严重的安全问题之一。2011 年以来的密码泄露事件令人触目惊心[68]。首先传出CSDN 遭遇黑客攻击，600万用户帐号及明文密码泄露，用户资料被大量传播。紧接着不久乌云漏洞平台再次爆出天涯社区4000万用户资料泄露，用户明文密码泄露。一旦恶意用户获取了合法用户口令，便可轻易绕过身份认证，进入系统进行信息窃取，进而不留痕迹的利用被攻击者的账号权限实施攻击破坏。正因为口令和智能卡的易丢失性，特别在内网环境下，被冒用的机会更高，因此这两种方法在内部网络环境下面临着较大的挑战。

3.1.1.2　基于生物生理特征的身份认证技术

生物特征识别技术是指通过人们的生理特点对用户进行身份鉴别。该技术不仅可以用于对当前使用者进行身份认证，同时也可以采集攻击者的个人生理特征信息，可用作身份分析和追踪，因此在访问控制和身份认证领域被广泛应用。由于每个个体的生物特征都是独有的，几乎不可能被轻易窃取或模仿，因此该类方法比基于口令的认证方法和基于智能卡的认证方法更可靠除此之外，用户也不必再费尽心思的记住冗长的密码或担心将认证物件遗落在家了。其主要缺点就是在对用户相关行为信息进行采集的同时，可能会触及用户隐私，会引起用户的相关担忧。通常被用作认证的生理特征包括以下几种。

- 指纹认证：指纹是手指末端指腹上由凹凸的皮肤所形成的纹路，也可指这些纹路在物体上印下的印痕。纹路的细节特征点有起点、终点、结合点和分叉点。由于每个人的指纹并不相同，同一人的不同手指的指纹也不一样，指纹识别就是通过比较这些细节特征的区别来进行鉴别。指纹由于具有个体差异性及稳定性，早在中国古代便用于身份确认，当时人们以指纹或手印画押。在西方，1890 年代以后警察逐渐将指纹作为辨认罪犯的方法之一。1960 年代随着电脑技术的发展，美国联邦调查局和法国巴黎警察局等开始研究电脑指纹识别技术。1990年代用于个人身份鉴别的自动指纹识别系统开发完成并推广应用。作为身份认证手段，指纹认证对手指的清洁度、湿度等都很敏感，油脂和外部伤痕等因素也会影响识别准确度。除此之外，现有研究还指出了某些人或某些群体的指纹特征相对较少，难以用来进行身份认证。

- 面部识别[67, 69, 70]：分析比较人脸视觉特征信息进行身份鉴别的一种计算机技术。面部的视觉特征是对人类（也包括其他动物）进行个体识别时的一种生物特性，个体之间的细微差别存在在自然界具有普遍性，这些细微差别能够被计算机系统捕获和识别，用作身份鉴别。虽然人脸识别具有很多优势，但也存在很多困难，包括不同个体之间的差别细微，人脸的结构特征是基本一样的，需要比较准

确的算法来鉴别。另外人脸的外形受人的情绪、年纪和检测环境等情况影响很大，人可以通过脸部变化产生很多表情，且不同的角度，不同的光照对计算机系统看到的人脸特征有很大影响。

- 虹膜扫描[71]：与眼睛相关的生物特征较少受到环境的影响，虹膜纹理可以可靠地提供认证独特性支撑。虹膜是人眼瞳孔和眼白之间的环状组织，是人眼的可视部分，其位于巩膜和瞳孔之间，包含了最丰富的纹理信息，占据整个眼睛外观的65%。虹膜的形成由遗传基因决定，是稳定的身份鉴别基础。虹膜扫描需要使用照相机和辅助光线对虹膜进行扫描，比较不方便。

- 掌型比对[71]：可以通过手掌的骨骼、经络的形状大小等生物特征来识别用户的身份。扫描人手掌的骨骼经络需要类似于x射线扫描的专用设备支持，成本较高。

以上基于生物生理特征的身份认证技术同样具有很好的实用效果，但同样一般仅用于单次认证，不能满足内部威胁中要求的实时持续认证的需求。

3.1.1.3 基于生物行为特征的身份认证技术

生物行为特征不同于生物生理特征，不采用人的器官、外表等生理特征作为认证的基础，而是试图捕获用户某些行为方面的特征来对用户进行认证。由于现在人与信息系统之间的交互多通过可视界面操作，人机交互行为非常广泛且普遍，使用人机交互HCI(Human-Computer Interaction) 行为特征作为用户身份标识技术成为被关注的焦点。人机交互行为特征主要分为两大类，击键动力学(Keystroke Dynamics) 和鼠标动力学(Mouse Dynamics)。

- 击键动力学：键盘曾是人与信息系统交互的主要方式，不同的人在击键的力度，节奏等习惯方面都有独特的特征。已有的研究[72–77]主要采用的特征包括击键的时间，击键之间的间隔时间等体现击键节奏的特征，不同的键之间的间隔又因为键盘布局而不一样，比如字母键与(a–z) 符号键('[',']',',',',.') 等间隔时间与字母键内部的间隔时间是不一样的，同样，CTRL，SHIFT，ALT等控制键与数据键的组合情况其击键时间、间隔时间等等都不一样，由此能引申出很多维的特征来表示特定用户独特的行为特征。已有的算法提取不同的特征，采用不同的算法模型来表征个人的使用行为模型。比如文献[72] 对17 名实验者，每个人输入大约1000 个单词的情况下，用击键的平均按键时间，不同键的转移时间，以及按键的延迟作为特征，实验得到大约5.5%的FP和5.0% 的FN。而文献[74]中采用的特征是3 字符的时间差作为训练特征，即连续三个字符之间的latency，将154 个用户作为测试对象，每个人输入683个字符，实验得到了4.0%的FP和0.01%的FN。文献[76]在考虑击键行为的时候，也考虑了相关的应用信息，如用户在不同的应

用，PowerPoint,Word,Yahoo Messager 和IE 下的击键行为可能是不一样的。文献[77]采用了N-GRAPH 方法，两个输入键序列的距离等于对应位置的N-GRAPH之间距离的和，序列间的距离用最大的相同N-GRAPH 出现次数作为归一化处理。在其实验中，40 个用户分别输入两个包含300 个字符的文本，训练出合法用户行为模型。另外90个测试用户输入两个文本中的第二个文本。比较任意测试用户的输入S和用户模板，如果S和某用户的模板之间的平均距离最小，则认为是该用户产生的，其实验取得了FN=5.36%，FP=0的效果。

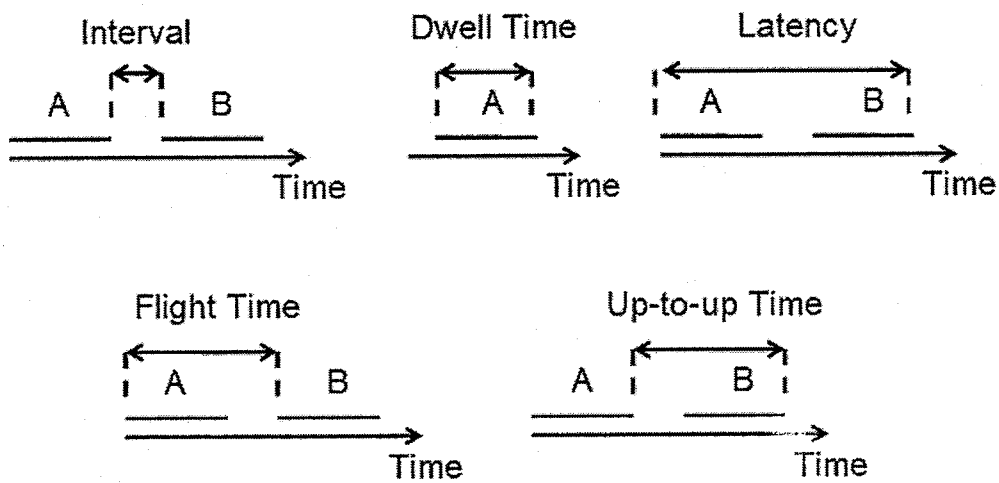


图 3.3: 击键动力学的时间间隔特征

- 鼠标动力学：尽管击键动力学研究的比较早而且准确性都比较高，但现代人机交互的形式已经越来越朝多样化方面发展，比如更多的鼠标操作，甚至用手势或者其他方式，键盘在日常的交互过程中参与度越来越低。鼠标动力学指根据人机交互过程中的人控制鼠标来移动光标所产生的行为特征进行身份识别的方法。鼠标在移动过程中所产生的特征，比如速度、距离、角度等都能构成了用户的唯一性特征。例如文献[78, 79] 中，Ahmed 和Traor 提出用神经网络模型对鼠标行为进行建模，他们的方法获得了2.46%的错误接受率FAR(False Acceptation Rate)和2.46% 的错误拒绝率FRR (False Rejection Rate)。在实验中，他们对参与实验的22个用户进行了284 个小时的实验，采集了998 个会话的数据。从原始的光标移动数据中，抽取了如鼠标移动速度距离比MSD(Mouse Speed Compared to Distance)，鼠标方向移动速度MDA(Average movement speed) 以及每种类型动作的平均移动速度ATA(Average Movement Speed per Types of Actions) 作为训练和认证特征。在他们的实验中，为了采集足够的数据（1000次有效的移动），一个会话大约要持续17-30 分钟。Pusara 和Brodley在文献[48] 中试图通过在一个时间窗

口内鼠标移动的夹角、速度和距离等特征来标志一个用户身份，他们采用C5.0决策树模型来进行分类，实验取得了0.43%的FAR和1.75% 的FRR。但是他们的场景是特定应用相关的。Gamboa和Fred 在文献[80, 81] 中提出了一种统计模式识别的方法来计算鼠标的移动模型，对一种web 游戏场景中的用户进行身份实时认证。他们开发了一种序列分类器来处理交互过程中的数据，按照该分类器，如果达到一定的准确性阈值，该用户身份被认为是正常的，否则作为骗子对待。实验采集了大约50个用户10个小时的数据，结果得到了2%的ERR($FAR + FRR$)。他们的实验环境同样是应用场景依赖的。

3.1.1.4 鼠标动力学的优势与不足

随着在人机交互中，鼠标的应用越来越普遍，而且人与人之间在控制鼠标行为方向确实有独特的个性，因此鼠标动力学更受到研究者们的青睐。虽然已有的研究成果声称了很高的FRR 和FAR，但是这些工作存在一些非常致命的弱点。文献[82] 对针对鼠标动力学做身份认证的几种方法做了评价，指出了已经存在的方法的几个缺陷，包括1)不切实际的认证时间.已经提到的认证方法大部分都需要十几分钟到半个小时的认证时间以便获取足够的认证数据，而这个缺陷在内部威胁场景下显得尤为突出，因为一个身份冒用者的攻击很可能几分钟甚至几秒钟就完成了，太长的认证时间根本来不及捕获数据，更检测不出恶意的身份冒用者；2）没有考虑实验中环境变量的差异性，比如鼠标的灵敏性，屏幕的分辨率等，每个用户的设置很可能是不一样的。Jorgensen等人重复了文献[48, 78] 所提到的方法，并严格在相同的环境变量（相同的操作系统，鼠标设备，显示设备，相同的系统设置）情况下，认证的错误率达到30%-40%，远超原作者所声称的不到5% 的错误率。

本文的工作正是基于前人在鼠标动力学方面研究的不足，提出了一套基于介入式场景设计的鼠标动力学身份认证系统，针对介入式的场景提取了许多有针对性的特征并通过实验证明了本系统的有效性。

3.1.2 介入式场景设计与系统实现

为了解决鼠标动力学身份认证技术的实用性问题，本节提出了一种在用户正常操作过程中插入短时介入式场景的思路，希望能在短时间内捕获用户的下意识鼠标移动行为，对当前用户身份进行持续认证。

3.1.2.1 介入式场景的概念

介入式场景的提出是希望能够在短时间内有效地捕获人们的鼠标行为特征。阅读文献[48, 78]发现过长的认证时间主要是需要过长的时间去收集足够的用户行为数据，因为系统无法把握当前用户的行为，操作用户可能随时会停住了鼠标移动，阅读显示

的内容或者通过击键来与系统交互。以前的研究者只有设计通用的采集器，采集到足够可以做认证的数据才会停止。过长的认证时间在检测威胁时存在巨大的缺陷，因为攻击者可能在很短的时间已经完成了攻击，而认证系统还未完成认证。为了克服这个问题，我们首先设计了几类介入式场景，然后再设计身份认证的关键点作为介入式场景的激活点。

1. 短时的介入式场景设计。介入式场景是指认证程序在用户使用电脑的过程中注入一些特定事件，引导用户进入一种短时认证场景。在这种场景下，有意让用户无法控制或者完全控制光标完成正常的移动操作，因此使得焦虑的攻击者急于想要快速找回对鼠标的控制，从而产生一些下意识的鼠标操作，比如快速来回移动。下意识的鼠标操作会产生不同人在焦虑情绪下的唯一性行为特征。

2. 身份认证关键点设计。身份冒用攻击者不会对通常的文件或者数据感兴趣。他们想要窃取敏感的隐私数据或者企业的机密数据，因此介入式场景的注入可以与被保护的对象关联起来。比如保护私人文件夹或者私人文件，当被保护的文件夹或者文件被打开时，介入式场景被启动。另外，应用程序也可以被分为敏感和非敏感的，比如电子邮件客户端，Outlook，Foxmail等被认为是私密的应用，当Outlook，Foxmail等应用程序被打开或者重新被从后台运行激活到前台时，介入式场景被激活。

以上两个思路从三个角度解决传统鼠标动力学的不足：1) 关键点的选择能够捕获用户有效的鼠标控制操作；2) 介入式场景下能够捕获到用户急躁潜意识下的大量数据；3）数据采集和认证的时间可以缩短。

基于上述讨论，3类介入式场景被设计，分别是光标停止场景(Cursor-Stopping Scenario)，光标消失场景(C ursor-Disappearance Scenario)，光标迟缓场景(Cursor-Slowing Scenario)。三类场景具体的实现策略包括以下几个步骤：

1. 创建一个Windows的透明窗口（MouseDynamicsWin）来遮住当前系统的前台窗口，并成为当前的前台窗口。比如当前用户打开了Outlook，进入介入式场景时，MouseDynamicsWin 成为当前系统的前台窗口。此时用户看到的还是Outlook 的操作界面。

2. 在MouseDynamicsWin中调用Windows SDK 的API 函数::ShowCursor(False)隐藏光标。由于窗口MouseDynamicsWin 为当前窗口，所以，其捕获了光标的焦点，能够控制光标是否可见以及光标的显示特性。

3. 设置各个场景下的控制策略。在光标停止场景中，程序获取原来光标的位置，在该位置画一个跟系统光标图形一样的光标图形，然后什么也不干，使得鼠标的移动无法在屏幕上显示为光标的移动。在光标消失场景中，什么也不做。在光标迟

缓场景中，实时捕获鼠标的移动事件，并根据当前光标的坐标和移动后的坐标计算坐标移动的方向和距离，将距离减半或者缩写为原来的1/3（一个预设的值）来计算场景下光标应该出现的位置，然后再该位置画一个跟系统光标图形一样的光标图形。

上述策略的关键是设置一个透明的窗口在前台，可以捕获鼠标的焦点，进而控制光标显示与否或者其应该显示的位置。在三类预设的介入式场景下，两个假设被提出：

假设3.1. 不同用户行为可区分性假设
在介入式的场景下，不同用户的下意识行为是可区分的；相同的用户下意识行为具有一定的一致性。

假设3.2. 不同场景可区分性假设
在介入式的场景下，相同的用户在不同场景下的行为也有一定的区分度。

在假设3.1的前提下，保证了介入式场景下用户的行为是可以区分的，而相同用户的行为是具有一致性的特征；假设3.2则保证同一用户在不同场景下的行为也是可区分的，这表明用户的正常行为和介入式场景下的行为具有可区分性，不同的介入式场景下的行为也具有可区分性。为了验证以上假设，本文设计了几个实验，画出不同情况下的用户的轨迹，如图3.4 和图3.5 所示：

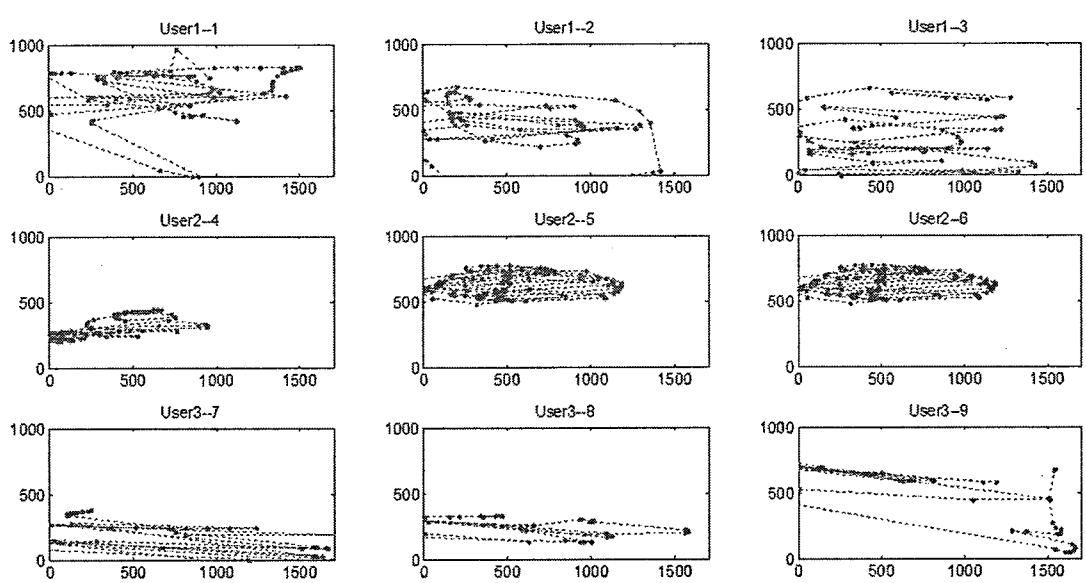


图 3.4: 不同用户在Cursor-Stopping场景下的移动轨迹图

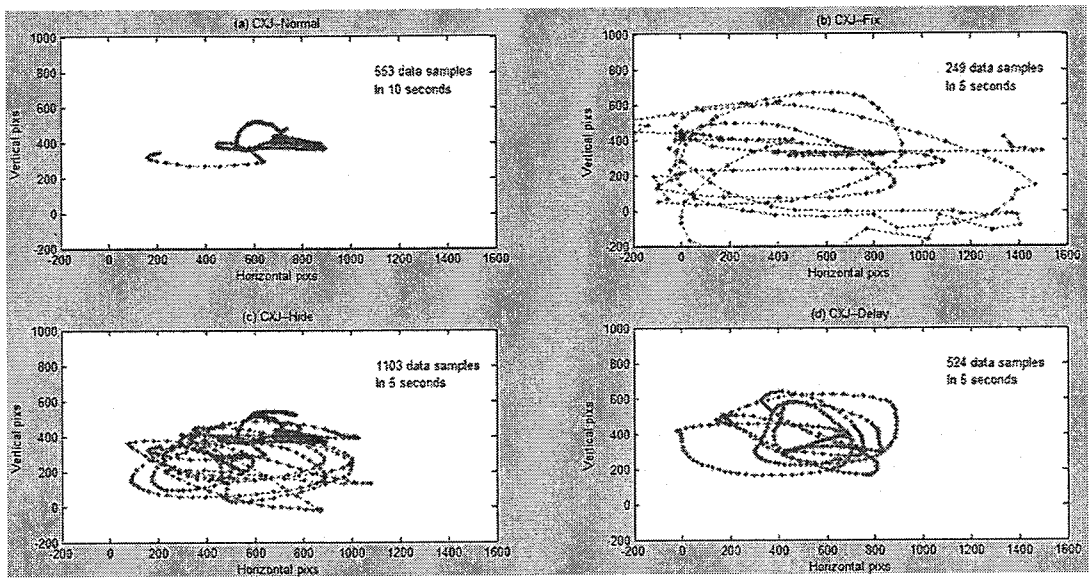


图 3.5: 相同用户在不同的场景下的移动轨迹图

图3.4表现了三个用户在光标停止场景下三次实验下的移动轨迹，横排的三个图代表同一个用户的三次实验。从图中可以看出，在感觉到失去鼠标控制的前提下，用户1和用户3 的反应是将鼠标左右晃动，移动范围很大，但是用户3更习惯于保持光标在水平线上，因此在不同移动点上，Y 轴上变化的范围很小。而用户2 与用户1 和用户3不同，习惯将鼠标左右急促、小范围内的圆圈运动。因此其移动轨迹集中在某一个更小的区域内。图3.5表现了一个用户正常情况下的鼠标移动轨迹与三个不同场景下的移动轨迹的对比。在正常情况下，鼠标的移动是均匀的，移动距离分布在更下的范围内。在光标停止场景下，用户因为急于寻找鼠标的控制，大幅度地晃动鼠标，因此其折返的频率与跨度都是非常大的，移动速度也非常快。在光标消失场景下，用户的表现是在光标消失的位置急促地来回移动鼠标，因此，表现出来在一定范围内的往返移动。在光标迟缓场景下，用户感觉到光标的移动比正常的操作下反应要慢，用户并未完全失去对鼠标的控制，只是移动变得迟缓，因此，用户的光标移动操作没有表现出特别频繁的折返现象，而是倾向朝同一个方向持续移动，而且横轴和纵轴方向的距离跨度都不是太大。

通过实验表明，不同用户行为可区分性假设与不同场景可区分性假设是普遍存在的现象，基于此假设设计的场景和实现算法是具有直观的解释性。

3.1.2.2　系统设计与实现

基于以上介入式场景的设计以及激活点的选择设计，本文实现了一套实现基于鼠标动力学的可行身份认证系统PAITS(Practical Authentication with Identity Tracking

System)，可根据鼠标行为对当前用户身份进行判定，并在系统中记录可疑用户的操作行为。PAITS系统分为四个组成部分，系统的体系结构如图3.6：

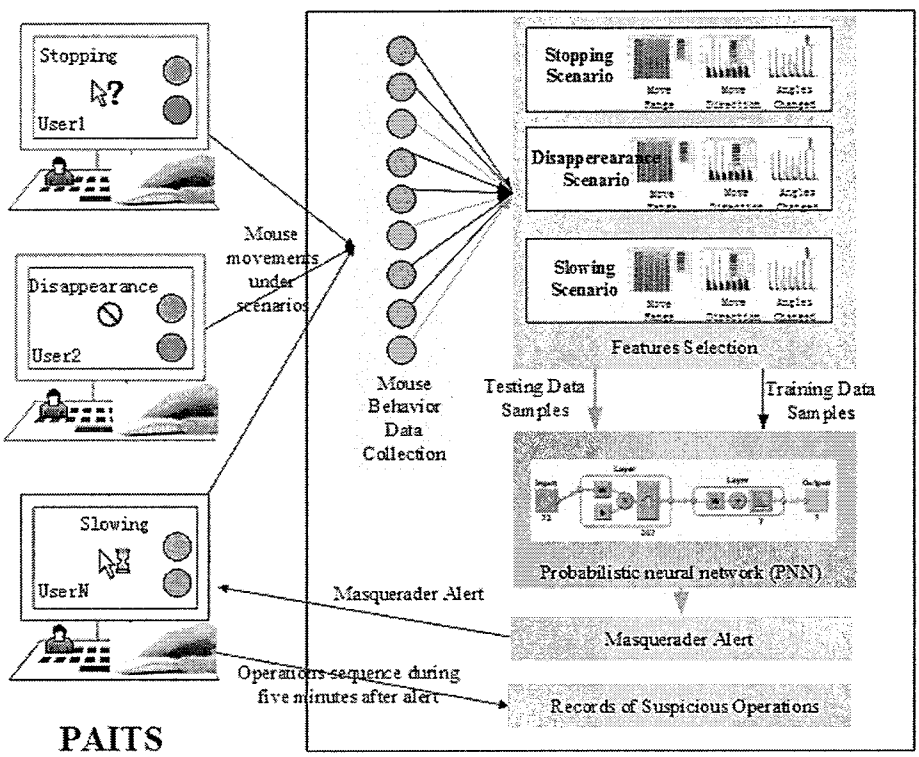


图 3.6: PAITS系统设计图

- 数据采集模块：数据采集模块实现了三类场景以及场景的触发机制。如前所述，场景的触发是在监控到某些敏感的文件或者文件夹被访问，或者重要的应用程序（如Outlook、QQ等应用）被打开时。场景的触发规则是可以配置的，系统提供了相关的配置文件Trigger.ini，规则的配置如图3.7 所示。规则可以是正则表达式，数据采集模块监控前台进程的运行情况，匹配规则，决定是否进入介入式场景采集数据。如果命中规则，随机选择一个场景激活，启动数据采集会话，会话时间预设为5 秒。

- 特征抽取模块：原始的鼠标移动坐标采样点被传递给特征抽取模块。特征抽取模块将一个会话的原始数据进行处理，从鼠标移动的范围、方向、速度和不同移动向量之间的夹角抽取合适的特征向量。

- 训练与检测模块：训练与检测模块工作在两个阶段。在训练状态阶段，根据特征抽取模块提取的特征向量针对特定用户的行为模型进行训练。在检测阶段，根据

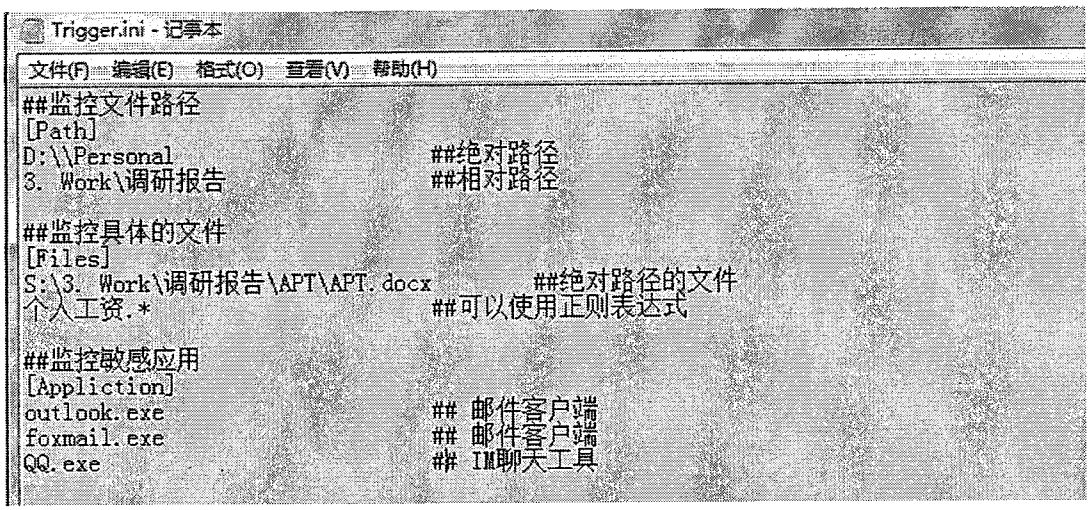


图 3.7: PAITS中触发认证规则的配置文件

已经训练好的模型检测会话的特征向量，决定当前会话是否是恶意用户或者正常用户的操作。

- 可疑操作追踪模块：如果恶意用户操作会话被检测出来，相应的可疑操作追踪模块会被启动。该模块发送一条操作监控命令到数据采集模块，数据采集模块接收到命令后，启动ProcessMonitor对当前系统的进程活动、I/O操作和网络应用情况进行记录，记录时间为5 分钟。操作记录会被数据采集模块发送给可疑操作追踪模块存储和管理。

3.1.3　特征提取与检测模型

本节主要讨论上述特定场景下的特征选择和检测模型问题。已有的研究主要提取移动的距离、速度、方向、夹角等作为特征集。如下图3.8 和图3.9所示的鼠标移动方向和鼠标移动夹角变化。通过这些移动的距离、速度、方向和夹角等的分布特征之间进行正交，通常会派生出几十个到上百个特征向量。研究人员再利用机器学习或者统计学习模型（如C5.0决策树模型、SVM 支持向量机模型和统计序列模型）对特征数据进行训练和建模。

本节针对介入式场景下采集的数据进行了详细分析，提取了71维数据特征，如表3.1所示。这些特征被分为三大组，鼠标移动的范围特征、鼠标移动的方向和速度特征和鼠标移动的夹角变化特征，下面将依次介绍。

3.1.3.1　鼠标移动范围的特征提取

图3.4给人的第一印象是：不同的人之间鼠标移动的范围是可区分的。为了刻画不

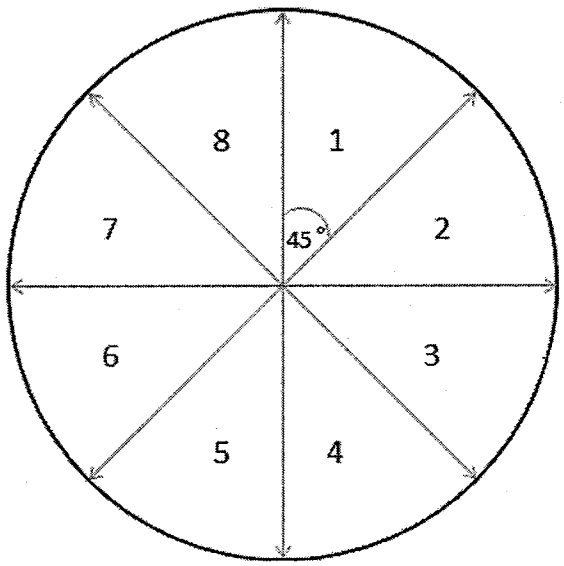


图 3.8: 已有研究的鼠标移动方向特征

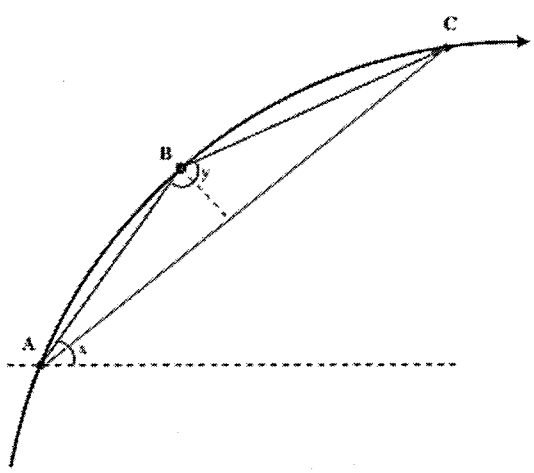


图 3.9: 已有研究的鼠标移动夹角特征

同人的特征，本节首先定义了两个值X_Range 和Y_Range，分别代表一次会话过程中鼠标移动在x轴和y轴移动的最大区间范围，具体计算如下：

$$X\_Range = max(point.x) - min(point.x);$$
$$Y\_Range = max(point.y) - min(point.y)。$$

$XRange$ 和Y_Range表示了在一次场景会话中用户光标移动的范围大小，比较直观地体现用户鼠标移动的幅度。

除了移动范围，在一个场景会话中用户的移动距离也各不相同。从表3.3中看到，

表 3.1: 71维特征表

序号	特征种类	特征的描述
1	范围特征	X_Range. x轴坐标最大值- x轴坐标最小值
2		x轴坐标的均值
3		Y_Range. y轴坐标最大值- y轴坐标最小值
4		y轴坐标的均值
5-12		x轴各区间的距离分布，分割点为[-∞, 200, 400, 600, 800, 1000, 1200, 1400, +∞]
13-20		x轴各区间的频率分布，分割点同上
21-26		y轴各区间的距离分布，分割点为[-∞, 200, 400, 600, 800, 1000, +∞]
27-32		y轴各区间的频率分布，分割点为
33	8个方向的距离、速度、频率的分布特征	方向I上移动距离的均值
34-40		方向I上移动距离的分布，分割点为[0, 25, 50, 100, 200, 400, 800, 3200]pixels
41		方向II上移动距离的均值
42-46		方向II上移动距离的分布，分割点为[0, 10, 20, 40, 80, 1280] pixels
47		方向I上移动速度的均值
48-52		方向I上移动速度的分布，分割点为[0, 1000, 2000, 4000, 8000, 32000] pixels/s
53		方向II上，移动速度的均值
54-57		方向II上，移动速度的分布，分割点为[0, 400, 800, 1600, 6400] pixels/s
58-65		移动在8个方向出现次数的频率分布
66-71	角度特征	两次移动向量夹角的角度分布，区间为0-180度6等分

大部分人的会话在X轴方向的移动范围都大于其在Y轴方向的范围，这与日常使用的显示器为宽屏的现象是相一致。其次大部分用户在一次会话的移动距离总共超过10,000像素点，比如User4总共移动了35,844个像素点，User6 总共移动了20,142 个像素点，也有少量用户移动的比较少，如User2 仅移动7,507 个像素，User3也仅移动了8,548 个像素点。图3.10 用直方图直观地体现了这种对比程度。

除了在移动的范围和距离有宏观的区分外，不同用户的移动区间也有不同的偏好。为了刻画这种区别，鼠标移动的空间做了如图3.11所示的划分。x轴上，划分为[0,200),

表 3.2: 不同用户的平均会话移动范围和距离表。单位：Pixels

User	X-Ranges	Y-Ranges	Distance
User1	$1.3626e+03$	794.7000	$1.4870e+04$
User2	924.8056	553.8889	$7.5070e+03$
User3	$1.2827e+03$	449.0982	$8.5484e+03$
User4	$1.9973e+03$	753.5444	$3.5844e+04$
User5	$2.0403e+03$	756.4342	$2.3708e+04$
User6	$1.8674e+03$	794.7460	$2.0142e+04$
User7	$1.3649e+03$	732.8571	$1.7393e+04$

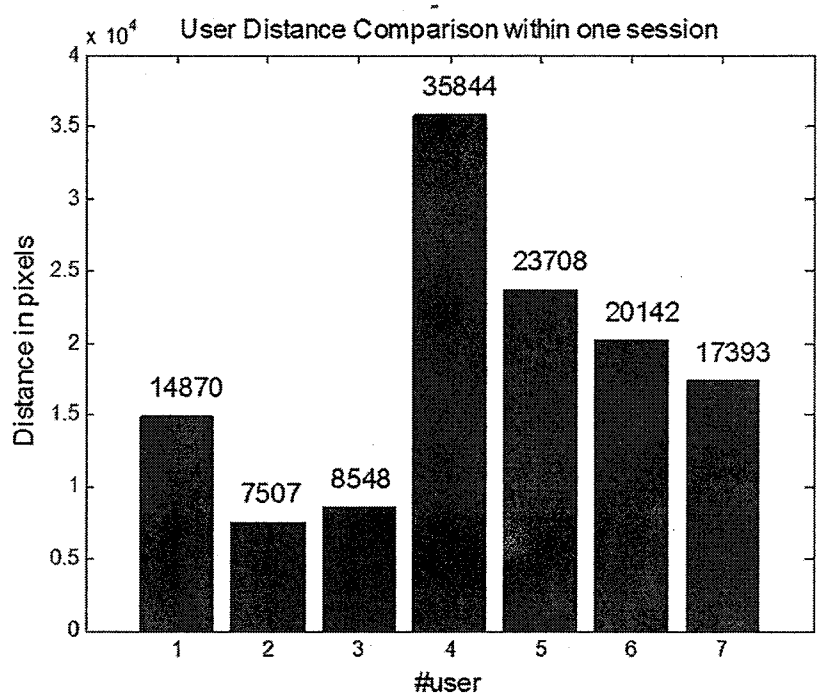


图 3.10: 不同用户的平均会话移动距离对比图

[200,400), [400,600), [600,800), [800,1000), [1000,1200), [1200,1400), [1400,+∞) 等八个区间的。y轴上，划分为[0,200), [200,400), [400,600), [600,800), [800,1000), [1000,+∞) 等六个区间。图3.12 展示了不同用户在不同区间移动的距离和发生频次的分布。

图3.12上面的两幅图表示用户在x轴区间上移动的距离和发生的频率分布，第一幅图表示在X轴各区间上移动距离的分布，第二幅图表示在X轴各区间上移动频次的分

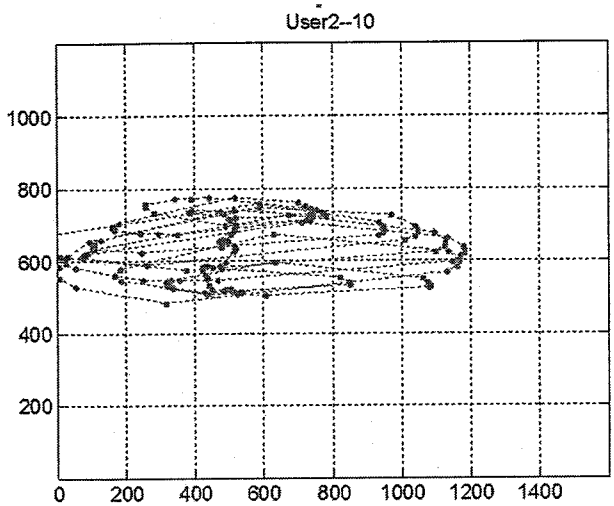


图 3.11: 移动区间划分示意图

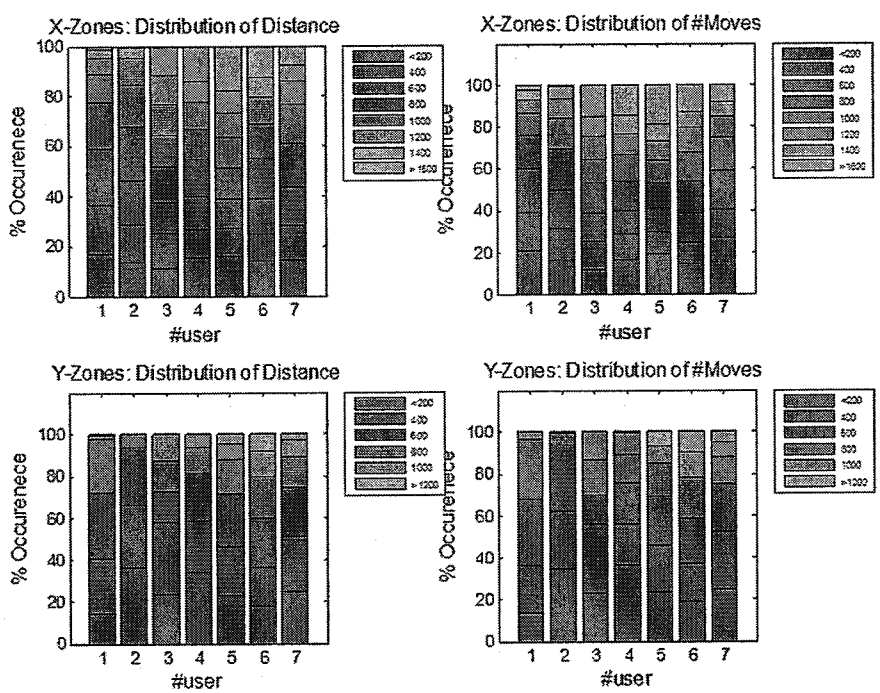


图 3.12: 不同用户在不同区间移动的距离和发生频次的分布

布。各用户大体上在X轴方向各区间的分布是比较平均的，其中User1 和User7 在接近屏幕两边的移动稍微少一些。下面的两幅图表示用户在Y轴区间上移动的距离和发生的频率分布，第一幅图表示在Y轴各区间上移动距离的分布，第二幅图表示在Y轴各区间上移动频次的分布。各用户在Y轴方向各区间的分布明显偏向屏幕的中上部，也就

是Y轴的中低位区间区，高位区间的移动频次明显很少，比如User2在1000以上的Y轴坐标区内移动的频次基本上不到5%。

3.1.3.2 鼠标移动的方向和速度特征提取

前面提到，前人的研究已经用了鼠标移动的方向这个特征，并将移动方向按照$\pi/4$为单位将所有移动方向划分为八个方向。本文也借鉴了这种方向的划分方法，考虑在八个方向上移动的距离、速度和频次的分布，分别编号为1–8。另外，在介入式场景下，八个方向上的分布导致特征维度过多，而用户的上下和左右移动特征比较明显，因此在统计距离和速度的分布时，传统的八个方向被聚为两类：方向I代表了1、4、5、8四个方向，可表示左右方向明显的移动；方向II代表了2、3、6、7四个方向，可表示上下方向明显的移动。方向划分如示意图3.13 所示。

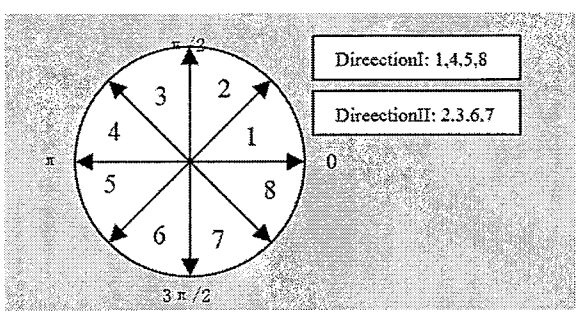


图 3.13: PAITS中鼠标移动方向的划分

表 3.3: 在各方向上的平均移动距离和移动速度(Pixels;Pixels/s)

User	Distance(I)	Speed(I)	Distance(II)	Speed(II)
User1	179.30	4156.63	37.74	1804.74
User2	95.48	2654.46	18.06	823.73
User3	160.31	2915.08	22.03	868.72
User4	371.31	10605.31	35.59	3096.64
User5	319.74	8203.18	46.11	3184.51
User6	224.22	21342.46	96.90	10961.58
User7	142.39	5209.28	64.45	3468.69

如表3.3和图3.14所示，不同用户在方向I和方向II上的距离速度分布情况各有不同。比如User4、User5主要表现在水平方向一次移动的平均移动距离为371.31,和319.74,

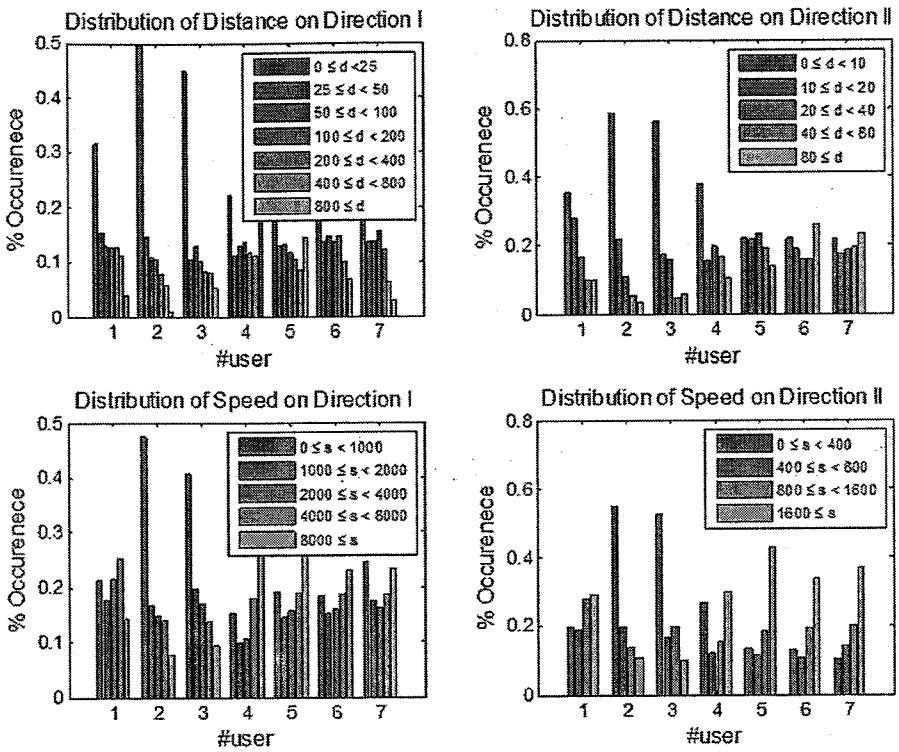


图 3.14: 不同用户在方向I和方向II上的距离和速度分布图

垂直方向上的平均移动距离为35.59和46.11，相应的速度也有同等比例的对比。而User6和User7在水平方向和垂直方向的对比不是那么强烈，其水平方向平均移动距离为224.22和142.39，而垂直方向的平均移动距离96.90和64.45。

3.1.3.3 鼠标移动的夹角变化特征提取

鼠标在移动过程中的方向变化也是值得考虑的特征之一。将鼠标的一次移动定义为两个采样点的坐标，即可计算该移动向量计算移动的方向，距离和速度等。定义两个连续移动之间方向的变化为这两个移动向量之间的夹角，其取值范围在$[0,\pi]$之间。将鼠标移动的夹角变化范围划分为6等分，分别为$[0,\pi/6)$, $[\pi/6,\pi/3)$, $[\pi/3,\pi/2)$, $[\pi/2,2\pi/3)$, $[2\pi/3,5\pi/6)$, $[5\pi/6,\pi)$，然后统计不同用户的移动夹角在不同区间内的统计。统计结果如图3.15所示。大部分用户的移动夹角分布在区间$[0,\pi/6)$,$[\pi/6,\pi/3)$和$[5\pi/6,\pi)$内，这也说明大部分人在场景下的行为都是急促地来回移动，但是其他三个区间的分布体现了不同用户移动轨迹的圆滑性。

3.1.3.4 概率神经网络模型

用来做用户行为建模和检测的学习算法有很多，已知的C5.0 决策树算法，支持向

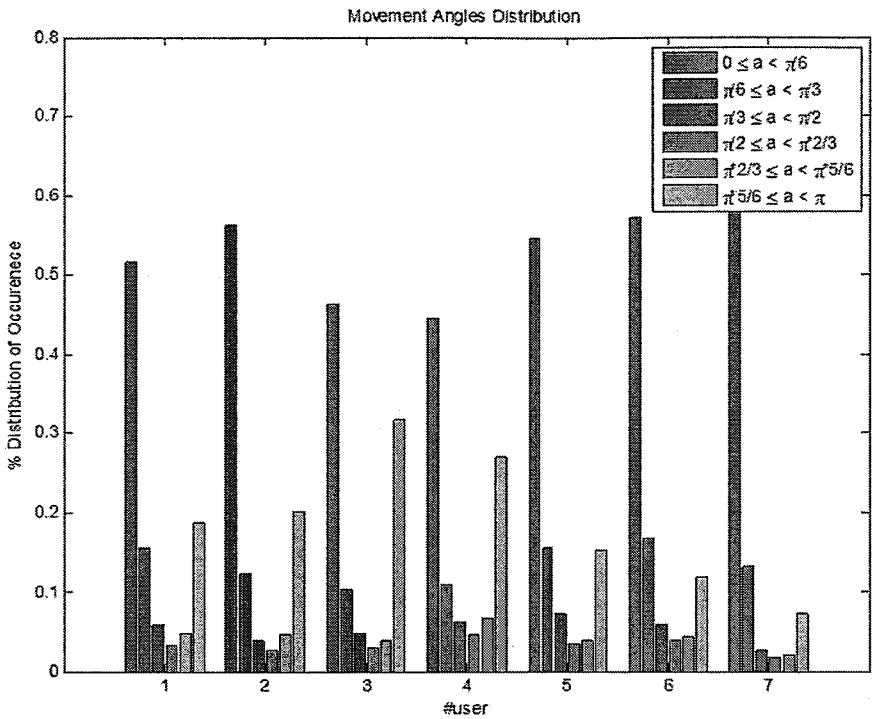


图 3.15: 不同用户在方向I和方向II上的距离和速度分布情况

量机SVM(Support Vector Machine)、神经网络(Neural Network)以及线性回归方法已经被人们广泛使用。在PAITS系统中，概率神经网络PNN(Probability Neural Network)被用来对鼠标移动轨迹行为建模。PNN是一个[8, 83, 84]后向反馈的神经网络，从贝叶斯网络和核化费舍尔判别分析方法KFDA(Kernel Fisher discriminant analysis)派生出来。PNN利用新的指数函数代替神经网络里面的Sigmod 函数作为神经元激活函数，整个网络分为四层，分别是输入层(Input Layer)，模式层(Pattern Layer)，综合层(Summation Layer)和输出层(Output Layer)，如图3.16 所示。和其他多层感知网络MPN （Multilayer percetron network）相比，PNN 有以下优点和不足：

- PNN模型比大部分MPN模型都要快的多。
- PNN模型也比大部分MPN模型准确。
- PNN模型对异常情况相对不敏感。
- PNN模型能够对测试用例生成比较准确的分类概率分值。
- 在分新的类样例时，PNN模型比其他MPN模型要慢；
- PNN模型在计算时要求更多内存空间。

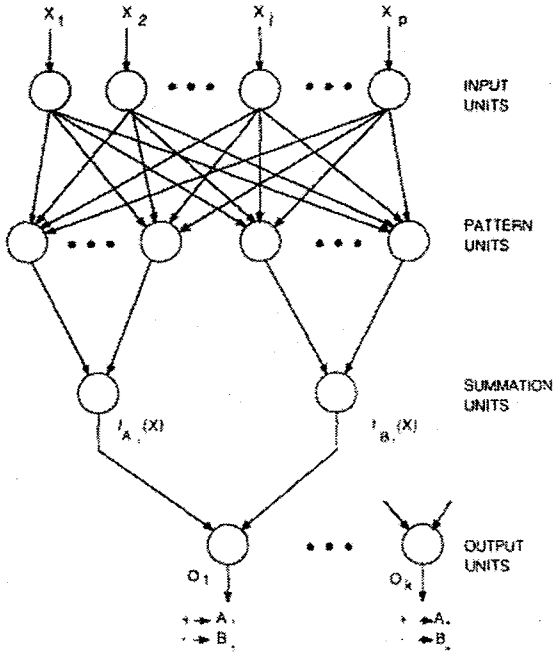


图 3.16: 概率神经网络分类映射示意图[8]

另外，为了评估PAITS系统认证的准确性，以下两个度量指标被定义：

FAR(False Acceptance Rate): 衡量认证系统将身份伪冒者错误地接受为正常用户的情况。FAR被定为所有身份冒用攻击会话中被认为是正常用户会话的比例。

FRR(False Rejection Rate): 衡量认证系统将正常用户错误地拒绝为身份伪冒者的情况。FRR被定为所有正常用户会话中被认为是身份冒用者会话的比例。

3.1.4 实验与结果分析

3.1.4.1 实验环境

本节实现了PAITS系统，并将数据采集模块安装到了12个用户的电脑上。这12个用户是实验室的正常用户，每天要用个人计算机处理正常的日常工作。数据采集器随着系统启动会自动运行，其数据采集过程对用户来说是透明的，除了在进入介入式场景后会对用户产生轻微的不适应感之外，对用户的操作没有任何影响。

如前所述，环境变量会对认证结果有巨大的影响。不同用户使用不同的硬件类型，操作系统或者有各自的系统设置，来自不同环境变量的数据所展现出来的差异有可能会影响实验的可信性。因此，实验环境应该尽可能调整一致，但是实际环境每个用的PC各有不同，完全要求所有的设备与设置都一样非常困难。在下面的实验中，所有用户的个人计算机环境可以分为以下四类，如表3.4所示。其

中，所有的操作系统都是Windows7或者Windows8，这两种操作系统在鼠标驱动和控制方面的差别并不大。鼠标设备类型总共有两种，包括lenovo MO28UOA 和lenovo MOEUUO，指针速度滑块设置都设置为一样的值：6/11，同时EPP功能(Enhance Pointer Precision)在Windows7和Windows8中都可被激活，用来增加鼠标移动的精度，所有的鼠标设备都启动了该功能。另外系统的其他设置如显示、字体大小都被设置为相同。

表 3.4: 所有实验用户的电脑配置情况表

类型	操作系统	鼠标设备	屏幕分辨率	光标速度	EPP
1	windows 7	lenovo MO28UOA	1280*800	OS default	OS default
2	windows 7	lenovo MO28UOA	1600*900	OS default	OS default
3	windows 8	lenovo MO28UOA	1600*900	OS default	OS default
4	windows 7	lenovo MOEUUO	1680*1050	OS default	OS default

3.1.4.2 实验数据与结果分析

实验过程中，12个志愿者被分为两组，其中7个为正常用户，5个为模拟的恶意用户。在训练阶段，7 个用户的鼠标移动会话数据被采集并分别为各个用户训练PNN模型。在测试阶段，5 个恶意用户间歇性地在7 个正常用户的电脑上使用工作，在这期间，12个用户的会话数据均被记录下来，并当作测试数据送到为7 个正常用户建立的PNN模型中。从12 个用户中采集到了总共1038个有效的会话数据，平均每个用户86个有效会话，其中757 个光标停止场景下的会话，144 个光标消失场景下的会话，137个光标迟缓场景下的会话。

在光标停止场景下，通过调整PNN模型的传播速度参数(Spread Value)，计算不同参数下的FAR和FRR，对应的ROC(Receiver Operating Characteristic Curve)曲线如图3.17所示。在将传播速度参数设置为0.36-0.38的时候，取得了最优实验结果，2.86%的FRR和4%的FAR。

3.1.5 小结与进一步讨论

这一节讨论了各种身份认证方法。基于口令和智能卡的认证方式最为普遍，但是存在很大的泄露和遗失风险。而基于生理特征的认证方式尽管可靠性非常高，但仅适用于首次认证，而且代价比较高，不太适合内部威胁要求实时认证的需求。基于生物行为特征的认证方式又包括击键动力学和鼠标动力学等方法。鼠标作为现代操作系统最重要且最流行的人机交互方式，因此基于鼠标动力学的认证技术非常有意义。但是

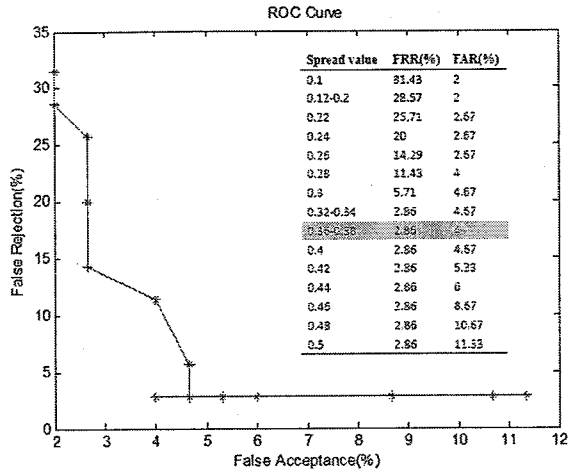


图 3.17: FAR与FRR的ROC曲线

已经提出的研究方法大多数不实用且没有考虑环境变量的影响。本节所提的介入式场景认证方法在攻击者不知情的情况下捕获当鼠标失去控制时，焦急下意识心态的鼠标操作行为，然后实现了一个PAITS系统，实现了数据捕获，特征抽取，认证过程以及可疑行为记录等功能。实验也证明PAITS能够在不太影响准确性FAR和FRR的情况下，将认证时间缩短到5 秒，极大提高了方法的可实用性。

介入式场景依然会影响用户的操作进程，虽然只有短短的5秒钟，如果攻击者意识到这一点，也许会有意识地避开数据监测从而使得数据采集模块获取不到相关的数据，认证过程失败。因此怎样提高介入式场景的隐蔽性是这个方向需要讨论的下一步工作。

3.2　内部威胁的异常行为检测

3.2.1　文件访问异常行为检测

文件作为数据在信息系统的基本存储方式，人对文件的访问行为是信息系统内人的基本行为。恶意的内部攻击者可能会通过搜索、阅读大量的文件而展示出与正常用户不同的特征，比如访问量、访问文件占文件系统中文件总数的比例大幅度上升等。部分恶意用户可能通过降低窃取文件的速度，延长时间等方法逃避常规的检测。据媒体披露[85]，斯诺登利用改进的爬虫软件，通过调整爬虫软件的速率、范围、周期从美国安全局窃取数据。本节集中讨论文件访问行为异常的内部威胁检测方法。

3.2.1.1　批量文件访问行为异常

内部攻击者窃取机密文件时很可能出现短时间大量文件访问的现象，称之为批量文件访问。考虑一个例子

例子3.1. *恶意用户Bob成窃取了Alice的文件服务器权限，利用Alice的账号登录文件服务器FileServer，利用grep等工具发现与某项目“XXXX”相关的项目文档，并下载到本地。*

考虑以上案例，Bob在其发起攻击期间会形成一次或者多次批量文件访问的行为会话。一次批量文件访问应该有如下几个特征：

- 访问会话时间短：可能只有几分钟甚至更少的时间；
- 访问的文件数量多：访问的文件数量多指各种文件操作接触过的文件的数量；异常示例如图3.18 所示；
- 文件类型访问比较广：恶意用户可能会访问各类机密文件类型，包括XLS/DOC/PDF/PPT等文本类型文档，也包括JPG/CAD/VSD等图形类型文档，也可能包括各种源代码文档，多媒体文档等等；
- 文件总量比较大：由于文件数量多，文件大小也没有限制，其总量可能会很大；
- 文件目录访问的广度和深度：普通的日常工作可能只会涉及到有限的文件夹和有限的深度搜索。而恶意的用户可能在文件目录的访问广度和深度都要多得多。

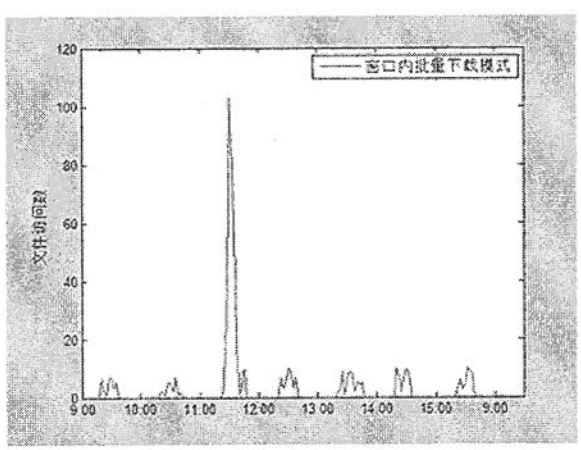


图 3.18: 内部攻击中文件访问数量异常示意图

根据上述特点，本节设计了在Linux服务器上的文件访问行为监控软件，这里的文件访问行为指利用文件系统I/O 监控所能捕获到的对文件的各种操作，包括List，Open，Read，Copy，Write，Remove，Close等操作。为了排除日常工作活动所带来的影响，把正常的工作活动中用的比较多的操作比如List，Write，Remove，Close等操作排除在外，只考虑Open，Read，Copy等操作。在文件访问行为监控软件中，一段时间窗口t内，一条文件访问行为特征向量被生成，特征向量的特征项如表3.5：

表 3.5: 文件访问行为特征表

特征	说明	特征	说明
FileNum	访问所有文件的数量	Flow	访问所有文件的字节数
ProjectNum	访问工程文件数量	ProjectFlow	访问工程文件的字节数
CodeNum	访问代码文件数量	CodeFlow	访问代码文件的字节数
MediaNum	访问多媒体文件数量	MediaFlow	访问多媒体文件的字节数
OtherNum	访问其他文件数量	OtherFlow	访问其他文件的字节数
FirstPathNum	访问一级目录个数	SecondPathNum	访问二级目录个数
ThirdPathNum	访问三级目录个数	#	#

批量文件访问行为异常的检测过程如下：首先对正常用户采集足够的训练数据E，$e \in E$表示一条数据记录，包括表3.5中的所有项，属性项i用e_i表示。对过去一个时间段用户的文件访问行为特征用$\hat{e} = mean(e), for\ all\ e \in E$ 表示。在检测当前窗口内的文件访问行为e'是否异常时，计算方法如下式所示：

$$Score = \sum_{i=1}^{n}(e'_i > \hat{e} : 0?1)/n$$

其中n表示特征的数量，在本方案中$n = 13$。定义一个阈值ω_1，如果$Score \neq \delta$，则认为当前窗口内的文件访问行为是异常的。

3.2.1.2　周期性文件下载行为异常

周期性下载模式主要用于针对恶意用户为逃避流量检测，而通过工具定期自动下载少量文件的方式，通过长期累积达到窃取大量机密信息的行为。周期性的下载行为示例可能如图3.2.1.2所示。 周期性文件访问能通过很多特征表现出来，本文采用文件访问时间间隔的方差和访问文件大小的方差来表示用户的在一个时间窗口的访问行为，然后通过定义不同窗口之间的相似性来度量一段时间范围内是否存在周期性行为，实现对周期性文件访问行为的检测。具体而言，首先定义同一个窗口内的周期性度量R为如下：

$$R = \frac{\alpha_1}{\delta_c + \frac{\alpha_2}{\delta_{interval}}}$$

其中δ_c表示过去一段时间每个窗口访问文件大小的方差，$\delta_{interval}$表示过去一段时间内每个窗口访问文件间隔的方差。α_1, α_2是可变的参数。R代表一个时间窗口内的访问情况，

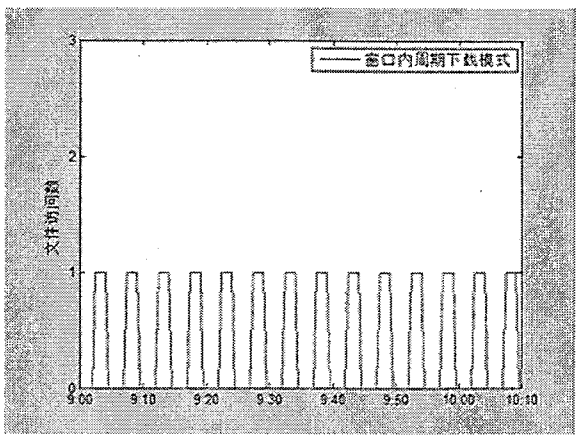


图 3.19: 内部攻击中周期性下载异常示意图

具有细微周期性的检测特征。对一段时间内每个窗口的访问会话的R进行直方图统计，如果某个直方图内落入的会话数超过阈值ω_2，则认为该时间段具有周期性文件访问的行为。

3.2.2 内部木马心跳行为检测

网络失泄密的很大一部分泄密来自于窃密木马攻击，检测窃密木马对检测失泄密行为具有积极意义。大多数窃密木马控制端为了监控受控端的状态，两者直接会适时地通信，这种保活措施称为心跳行为，产生的数据包称为心跳数据包。这些数据包或者具有周期行为，或者为了躲避统计分析进行了随机化处理。也有些窃密木马也会使用TCP协议自身提供的心跳机制。

目前窃密木马主要的心跳行为类型如表3.6所示。

TCP Keep-alive机制心跳行为利用TCP协议自身提供的心跳机制，通信一端在空闲时向对端发送一个字节的数据，另一端返回TCP报文。TCP连接内心跳行为指在一个TCP连接内部，通信一端周期性的发送固定长度的报文。TCP连接级心跳行为指木马程序每隔一段时间想另一端发起TCP连接，连接成功后即断开连接，表现TCP短连接。

目前对木马心跳行为已有的检测方法主要有三种。第一种是基于规则的检测，例如将“存在连续多个大小相同的报文，且到达间隔时间差小于某个阈值”作为一条检测规则。该方法能检测某些木马心跳行为，却无法检测到心跳数据包随机到达的窃密木马，容易产生漏报。第二种方法是基于傅里叶变换的周期检测方法。该方法对包的到达时间间隔进行离散傅里叶变换，由于窃密木马心跳具有周期性特征，其高频系数接近于0，低频系数很大，而正常通信的低频系数与高频系数差异不如窃密木马的心跳行为大，因此可以利用高低频系数的差异值可以作为木马心跳行为的检测依据。该方

表 3.6: 窃密木马心跳行为说明

心跳类型	心跳说明	木马举例
TCP Keep-alive机制心跳	TCP协议支持的心跳类型，一端在连接空闲的时候发送一个字节，另一端返回ACK报文	灰鸽子
TCP连接内心跳	木马程序在一个TCP长连接内每隔一段时间发送固定长度的报文，另一端返回固定长度的响应报文	上兴远控木马
TCP连接级心跳	木马程序每隔一段时间向另一端发起连接，连接成功后断开连接，主要为TCP短连接	PCShare

法的主要缺点是只考虑了报文的到达时间，没有考虑报文大小，容易产生误报，而且该方法计算复杂，开销较大。第三种方法是基于小波分解的方法。该方法简化了傅里叶变换的计算，只计算高频系数，如果高频系数低于阈值，则认为是心跳行为。这种方法计算简单，但继承了傅里叶变换检测方法中容易产生误报的缺点。

基于此，为了准确地检测多种窃密木马心跳行为，本节提出了一种有效的窃密木马TCP心跳行为检测方法，主要包含三个步骤：

1. 网络数据包的抓取和TCP数据流的还原；

2. TCP心跳行为检测；

3. 误判检测。

网络数据包抓取和TCP数据流还原是数据采集和预处理过程，经过预处理后根据之前定义的三类TCP心跳行为特征进行心跳行为检测，最后有误判检测过程排除心跳检测过程中正常软件的心跳行为，使检测结果更加准确。

针对大多数窃密木马表现出来的心跳行为，基于网络数据包大小、方向和时间等特征，能计算出心跳的周期及其波动的范围。为了不干扰正常的网络通信，窃密木马心跳行为检测服务器捕获并分析从交换机上旁路过来的流量，通过监控TCP流发现窃密木马的心跳行为，并对可疑心跳行为进行报警。其实施的网络环境如图3.20所示。窃密木马心跳行为检测技术分为四个步骤

1. 网络数据抓取过程。实现一个网络流量sniffer，可以运行在网关处理。抓取网络数据包，还原TCP数据流，并记录TCP 流信息：通信开始时间BeginTime、结束时

间EndTime、源IP地址SIP、目的IP地址DIP、源端口SrcPort、目的端口DstPort、数据包的字节数PacketLength、数据包到达时间PacketTime、序列号Seq和确认序列号SeqAck。

2. TCP Keep-alive心跳行为检测。主要通过TCP流中客户端和服务端发送数据包长度和序列号进行判断。检测TCP Keep-alive心跳行为的条件为心跳数据包数量超过阈值$MinKeepaliveCount$。其中判断一个数据包是否为心跳数据包的条件如：

$$\begin{cases} SeqAck_c - Seq_s = 1 \\ PacketLength_s = 1 \end{cases} or \begin{cases} SeqAck_s - Seq_c = 1 \\ PacketLength_c = 1 \end{cases} \tag{3.1}$$

其中下标s和c分别表示数据包来自服务端和客户端。

3. TCP连接内心跳行为检测。根据一个连接内每个方向的数据包大小和时间，判断大小相似的数据包的发送序列是否具有周期性。如果周期性明显则可认为是可疑连接内心跳行为。检测TCP连接内的心跳行为，由于心跳报文大小比较小，需要先过滤掉大于$MaxPacketLength$ 的数据包，然后将大小相似的数据包聚为一类。相似度倒数ρ的计算公式见等式3.2：

$$\rho = \frac{PacketLength - PacketLength'}{\overline{PacketLength}} \tag{3.2}$$

其中$PacketLength$和$PacketLength'$分别表示两个不同数据包的大小，$\overline{PacketLength}$表示连接内平均包大小。ρ 越小，相似度越大。ρ 小于阈值ω的数据包被聚为一类。每一类中的数据包按到达时间排序，计算相邻数据包到达时间差值的均值和方差。判断该连接是否有心跳行为的条件为3：

$$\left\{ k \| \frac{\sum_i (T_{k,i+1} - T_{k,i})^2}{n_k} - \frac{(T_{k,n_k} - T_{k,0})^2}{n_k^2} \| < \delta,\ n_k > N \right\} \neq \varnothing$$

其中T_k, i表示第k个类中的第i个数据包的到达时间，第k个类中共有n_k个数据包，Δ是方差阈值，N是数据包数量阈值。数据包的聚类算法首先对原始集合中的数按大小排序，计算每个元素的相似元素的个数。然后按相似元素个数从多到小选取类中心点，该中心点与其相似元素构成一类。反复进行，直到原始集合全部元素最终被选完。

4. TCP连接级心跳行为检测。分析具有相同三元组(SIP，DIP，DstPort)的若干连续TCP短连接的时间和通信字节数，判断通信字节数相似的连接序列是否具有周期性。如果周期性明显则确认为可疑TCP连接级心跳行为。检测连接级的心跳行为，采用和TCP连接内心跳行为检测类似的算法，只是计算的是多个连续连接的通信数据，而不是一个连接内的通信数据，因此使用不同的到达时间方差阈值和数据包数量阈值，分别为Δ和N'。

上述每类心跳行为的检测过程最后都需要排除正常程序通信行为产生的心跳特征。判断是窃密木马心跳行为的条件为等式3.2.2：

$$\begin{cases} Sum_{out}/Sum_{in} > \varnothing \\ EndTime - BeginTime > MinConnectionTime \end{cases}$$

其中Sum_{out}和Sum_{in}表示排除心跳报文或连接后内网主机发送和接收的字节数，$\varnothing$是比值阈值，$MinConnectionTime$ 是窃密木马通信最小持续时间阈值。

3.2.2.1　实验结果及分析

在实验室环境3.20中，各参数设置如表3.7所示。运用该算法分析局域网多个TCP流，能够准确检测到PCShare、上兴木马远控和灰鸽子三种类型的心跳行为。图??分别给出了三种木马的检测结果。上面的图表示灰鸽子木马受控端在空闲的时候向控制端发送一个字节的心跳报文，为典型的TCP Keep-alive心跳。中间的图表示上兴远控木马受控端在一个TCP 连接内每隔30s发送3个字节的心跳报文，为TCP连接内心跳行为。最下面的图中PCShare控制端每隔13s 向受控端发起连接，并发送745字节的心跳报文，是典型的TCP连接级心跳。由实验结果可知，本方法检测结果准确，适合实时在线检测具有各种心跳行为的窃密木马。

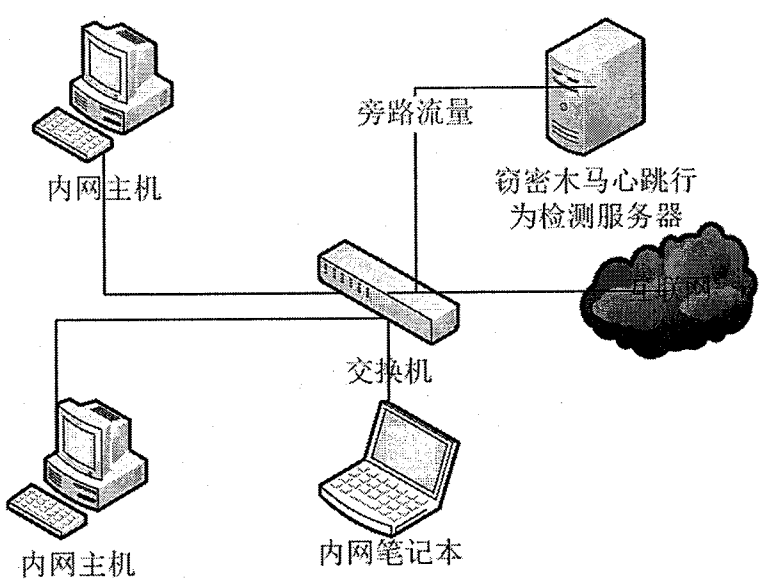


图 3.20: 内部木马心跳检测的环境图

3.2.3　小结与进一步讨论

本节讨论了在内部威胁中遇到的两类行为异常情况，分别是通过监控用户的文件

表 3.7: 木马心跳检测算法参数设置

参数名	参考值	参数名	参考值
$MinKeepaliveCount$	3	$MinConnectionTime$	60秒
$MaxPacketLength$	1460Bytes	δ	100
N	10	δ'	1000
N'	10	Ω	0
ϕ	1.0	$ConnectionTime$	30秒

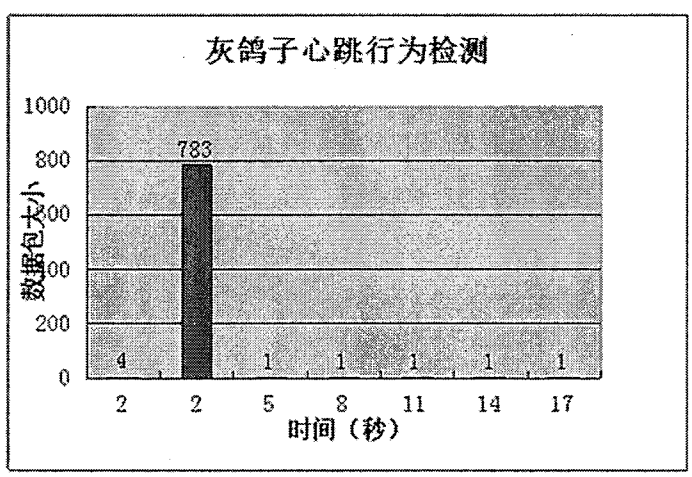


图 3.21: 灰鸽子木马心跳行为检测结果

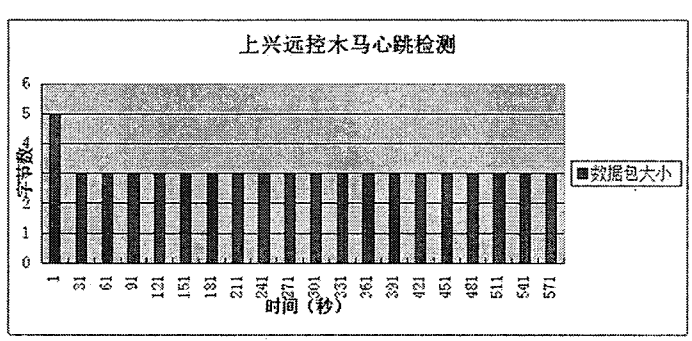


图 3.22: 上兴木马心跳行为检测结果

访问行为发现批量下载行为和窃密木马的周期性下载行为，通过监控网络通信流发现窃密木马的心跳行为。实验结果表明这些算法较好地发现相应的异常行为。

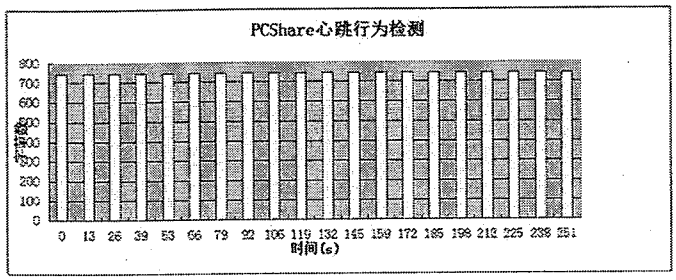


图 3.23: PCShare木马心跳行为检测结果

3.3　本章小结

本节讨论了内部威胁的异常行为感知技术研究，主要介绍了身份异常、文件访问行为异常和内部木马心跳行为异常等检测技术。这些异常检测技术针对性地检测不同的异常行为，给出异常告警和相关的参数，为后续的内部攻击意图理解和防护提供数据支撑。

第四章　面向内部攻击意图理解的概率攻击图模型

4.1　研究背景与问题概述

传统的内部威胁防护都是从单个关键节点监控去捕获特定异常行为，如相关工作中提到的通过数据库SQL访问对象的异常推断内部威胁，通过对系统的文件访问行为检测异常的内部攻击者，以及在电子医疗系统(EHRs)中检测医务人员非法访问病人病历等个人隐私的研究。这些系统通常只能以某种概率检测到用户的异常行为，而且对于过于复杂的攻击，比如潜伏期很长的多步攻击无法完整获得用户真正的攻击意图，对用户所采用的攻击路径一无所知，具有极大的局限性。具体而言，内部威胁提出了两个挑战：

- 内部攻击行为具有较强的伪装性，这导致检测结果具有一定的不确定性；
- 内部恶意人员可以有充裕的时间安排攻击行动，其攻击行为具有多步骤性和潜伏性，使得攻击行为很难被检测到。

内部攻击的伪装性体现在两个方面. 恶意内部人员可以滥用自己的合法权利，在正常行为遮掩下从事非法活动，称为行为伪装；也可以通过职便或者社会工程学的手段获取他人的身份认证信息，冒充合法用户进入他人电脑或数据中心窃取机密数据，称为身份伪装. 在具有行为伪装和身份伪装的内部攻击者面前，传统的防火墙和访问控制等手段基本会失效. 因此内部攻击行为检测方法更多地采用异常检测的算法来发现用户偏离正常行为模式的行为，从而确定攻击的存在，比如第一章提到的Schonlau等人[33] 通过检测用户的命令执行序列来发现异常行为，Salem等人[50] 对用户的文件搜索访问行为建立一类支持向量机模型来检测身份冒用者(Masquerader)，Zheng等人[86]通过鼠标移动等的生物行为特征来确定当前用户是否为合法用户。这些方法能检测一些异常行为，但也存在较高的假阳性FP(False Positive) 和假阴性FN(False Negative)，如Schonlau用贝叶斯一步马尔科夫模型检测执行命令序列，其FP 为6.7%，FN为21.3%。较高的假阳性会浪费网络安全管理员大量的人力分析成本，而错误的指控内部人员会带来破坏性的影响。

内部攻击的多步骤性体现在内部攻击者有足够的时间观察、计划和实施攻击方案，逐步攻击系统的弱点，达到攻击目标。这种多步骤性加重了内部攻击检测方法的复杂度。攻击树/攻击图模型可以模拟不同节点之间的因果依赖关系，常被人用来研究复杂的多步攻击行为。攻击图通过对攻击行为和目标网络进行建模，刻画了攻击者有步骤地实施攻击，一步步提升权限，获取更多的资源，并达到攻击目标的过程。

早期攻击图[7, 61]的构建用来表示攻击步骤之间的依赖性，建立网络系统的安全模型，发现攻击路径和可能导致攻击伤害的场景。比如Sheyner[7]首次提出了用五元组$< H, C, T, I, ids >$来描述目标网络，该模型对网络系统进行了合理的抽象，并且支持自动构建攻击图，对后面众多人的研究产生了深远的影响。Ou等人[56]进一步使用Horn逻辑来描述攻击模板，使得对攻击图的构建可以用逻辑描述模板的方式进行。图4.1描述了一个远程攻击的模板，表示主机上存在一个漏洞ID，该漏洞能允许攻击者提升访问权限，该漏洞可以通过网络接口Protocol，Port被访问，如果有人访问该网络接口，则可能该用户是一个恶意的攻击者。

```
execCode(Attacker,Host,Priv) :-
  vulExists(Host,VulID,Program),
  vulProperty(VulID,remoteExploit,
  privEscalation),
  networkService(Host,Program,
  Protocol,Port,Priv),
  netAccess(Attacker,Host,Protocol,Port),
  malicious(Attacker).
```

图 4.1: Horn逻辑语言描述远程内存溢出的攻击模板

在概率攻击图的推导计算方面，张少俊等人[59]提出了一个在满足观测事件偏序条件的情况下，利用贝叶斯推理计算攻击图节点的置信度方法。他们的模型中攻击节点决定了观测事件被观测到的概率，而不是通过观测节点推断攻击节点发生的概率，有一点不太好理解。另外该模型根据当前攻击节点的概率，使用似然抽样算法推导观测事件的置信度，有一定的局限性。在Wang 等人工作[57] 的基础上，叶云等人[60] 提出了一种攻击图简化算法来解决存在环时攻击图推导存在的性能瓶颈问题，基于简化的攻击图，给出计算最大可达概率的方法，他们的算法计算了最大可达概率，给出了一种意图推断的思路，但没有计算攻击场景。另外与Wang[57] 的工作一样，该工作也未引入观测事件的置信度进行概率计算。

攻击图的研究工作较好地刻画了多步攻击，但大都关注在预先固定的网络配置环境下的安全风险评估计算，没有利用当前攻击动作检测结果的不确定性进行概率计算，不能全面模拟内部威胁场景中的各种不确定性，在进行内部攻击意图的推断计算方面有一定的限制。而意图推断主要使用贝叶斯网络或者贝叶斯网络的变形（最大可达概率）来计算图上的概率推导，其基础的攻击图模型因为没有模拟完备的不确定性，从而导致上面的推导算法也不够适用于内部威胁。

4.2　内部攻击检测的不确定性

在一个内部的网络信息系统内部，攻击者发起一次网络攻击可能存在三种不确定性：

1. P1：攻击者的能力。当系统漏洞或管理漏洞存在，攻击者能够或者会利用漏洞发起攻击的能力。并不是所有的内部人员都会去攻击漏洞，也不是所有恶意的人员都有能力利用漏洞发起攻击。攻击者的能力体现在许多方面，见表4.1所示。这个概率属性和Dantu[87] 及Liu[88] 工作中引入概率含义是相同的，都刻画了攻击者自身能力的不同。

表 4.1: 不确定性一攻击者的能力体现

发动攻击的影响因素	解释
网络访问权限	攻击者是否有可能接触到被攻击目标网络，这决定了攻击者能否攻击目标网络的可能性
漏洞的难易程度	系统存在的漏洞的难易程度，这从客观上影响着攻击者是否有能力攻击该漏洞
内部攻击模式的难易程度	系统中存在一些内部攻击模式，比如利用sniffer侦听重要管理者的邮件。这种攻击模式的难易程度也从客观上影响攻击者是否有能力发起该攻击。
攻击者的专业技能	攻击者的专业知识从主观上影响着其是否有能力发起攻击。
攻击者的职业便利	某些攻击还要求攻击者具有一些职便等外部条件。比如“猜口令”攻击要求攻击者离受害人比较近，或者有一定的工作交集等。

2. P2：攻击发生的置信度。前面已经阐述过在检测内部攻击行为时，多用异常检测算法，比如身份冒用攻击，文件访问异常行为检测等。异常检测算法可能会有两种输出[89]，异常分值(Score)和异常的标签(Label)。给一个数据样本的异常分值取决于计算模型认为当前样本为异常的程度，因此这种技术通常允许分析师通过专业知识来调整阈值，选择最可能的异常样本点。基于标签的技术通常只给出样本点为正常或者是异常两种状态。相比较而言，基于异常分值的技术应用更为普遍。因此，当观测到异常事件时，对应的异常分值可以转化为意味着攻击发生的

概率，也就是攻击发生的置信度，本文用P2表示。传统的概率攻击图总是假定观测事件与攻击发生总是确定性的关系，但异常检测往往具有较大的假阳率FP。利用异常检测算法给出的异常分值作为攻击发生的置信度，这样描述观测事件与攻击发生的对应关系使得攻击图更合理且具有可扩展性.

3. P3：攻击成功的概率。特定步骤的攻击发生后，能够成功达到下一个状态的概率称为攻击成功的概率。攻击发生与攻击成功在很多场景下是相同的概念，因此也一直被混淆着使用，比如当检测到某个存在的远程调用溢出的漏洞可以获取主机的管理员权限，那么几乎可以断定攻击者已经获取了目标主机的管理员权限，因为这种检测会产生一种确定性的检测结果。但在内部威胁场景下，攻击发生并不一定预示着该攻击一定成功，下面的列表4.2列举了一些攻击发生与攻击成功不一致的例子。 攻击成功的概率与安全防护策略有直接的关系，加强安全防护策略能

表 4.2: 攻击发生与攻击成功不匹配的例子

内部攻击方式	攻击发生和攻击成功的关系
猜口令攻击	恶意用户试图进行口令攻击，系统出现了多次口令登录错误的日志，安全防护软件给出“猜口令”攻击的告警事件，但是该攻击能否成功取决于口令的强度和私密性
文件访问行为异常	恶意用户毫无目的地搜索某服务器上的文件夹，然而机密文档被存放在一个加密的磁盘中，比如以Truecrypt加密的虚拟磁盘卷中。此时，攻击虽然发生，然而却不能达到攻击目的。
数据库访问异常	恶意用户检索目标数据库中的大量数据，而敏感数据如用户名密码以较强的Hash值存放在某个字段里，此时，即使攻击发生，仍然不能成功。
身份冒用攻击	在第二章恶意用户利用正常用户外出而忘记锁屏的机会进入系统，此时身份冒用攻击已经成功发生，如果异常检测引擎在检测到异常行为时弹出二次密码认证窗口，能够及时阻止身份冒用攻击。

够影响攻击成功的概率。安全防护策略将在第六章讨论。

图4.2具体描述了3种不确定性的含义及其之间的关系。在攻击前，恶意用户发起攻击的概率由其攻击能力P1来刻画，此为攻击发生的先验概率；攻击发生后，如果观测事件被看到，那么攻击发生的后验概率被计算；攻击的效果由攻击发生的后验概率和攻击成功概率P3联合求解得到。

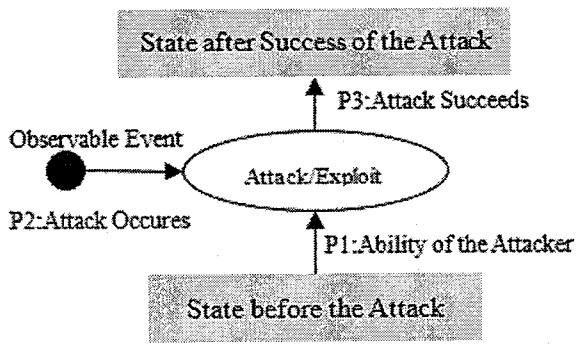


图 4.2: 攻击过程中的三种不确定性示意图

4.3 概率攻击图模型定义

内部攻击是一种很复杂的多步骤行为，包含着一些完成最终攻击意图所必须达到的子攻击目标，和一组紧密关联的基本攻击动作。基本攻击动作的实施帮助攻击者达到子攻击意图，攻击者相应地获得更多的资源和权限，以实施下一个次攻击。攻击图可以静态地评估一个网络系统的安全状况，但是很难动态地推断攻击意图，即根据当前安全系统的运行状态，评估可能存在的攻击。

为了建立适用于充满不确定性的内部攻击意图推断的模型，本文在Ou[56]的攻击图模式上进行修改，加入了三类条件概率转移表，以此来表示上述讨论的三类不确定性。概率攻击图的形式化定义如下：

定义4.1. 概率攻击图定义为有向无环图$PAG = (N, E, \Delta, \Gamma)$，其中：

- N代表节点集合，$N = S \cup A \cup O$。S是状态节点集合，每一个状态节点代表攻击者每步攻击后所处的状态，包括取得的攻击资源或权限能力。s_0是初始状态节点，表示攻击者最初所处的攻击状态。G 表示攻击目标节点集合，满足约束：$G \subset S$ 。A 是攻击动作节点集合，表示某个具体的攻击动作发生。而O为观察事件节点集合，假定每次攻击a_i 发生，安全监控系统能以一定的概率监控到某个事件o_i 。所有节点有一个概率属性，其取值范围为*(0,100]*，分别表示“攻击状态达到*(S)*”，“攻击意图成功*(G)*”，“攻击动作发生*(A)*”以及“攻击被检测系统观察到*(O)*”的置信度。。

- E是表示各类节点之间因果关系的有向边集合。具体E可表示为$E = E_s \cup E_a \cup E_o$ 。其中，$E_s \subseteq S \times A$ 表示攻击者拥有某些资源后才能实施某些攻击动作。$E_a \subseteq A \times S$ ，表示一次攻击成功后能够获取更多的资源，进入新的攻击状态。$E_o \subseteq O \times A$ ，表示根据安全监控系统观测到的事件可以确定或者推断某个攻击已经发生。

- Δ为条件概率表*CPT(Conditional Probability Table)*，依附于每一条有向边上。根据有向边的分类，$\Delta = (\Delta_s, \Delta_a, \Delta_o$，$\Delta_s$ 是依附于有向边集E_s上的条件概率表，比如$\delta_{s_{ij}}$ 表示在攻击状态s_i 下可能发生后续攻击a_j 的概率。Δ_a 是依附于有向边集E_a的条件概率表，$\delta_{a_{ij}}$ 表示攻击动作a_i成功并进入下一状态s_j 的概率。Δ_o 是依附于有向边集E_o 的条件概率表，δ_{o_i}表示观测事件的置信概率，观察到o_i事件时能证明攻击a_i 发生的概率，即：$P(a_i|o_i)$。

图4.3展示了攻击图的几种可能的情况。其中不带线条的圆圈且标记为$s_{0,1,2,...}$的节点表示攻击状态节点，攻击动作节点用带空心的黑色圆圈表示，且用$a_{1,2,...}$标记，观察事件节点用黑色实心节点表示，且上方带有标记$o_{0,1,2,...}$。每一条边都依附着一个概率属性，用灰色实体方框表示，且用$\delta_{1,2,...}$标记。

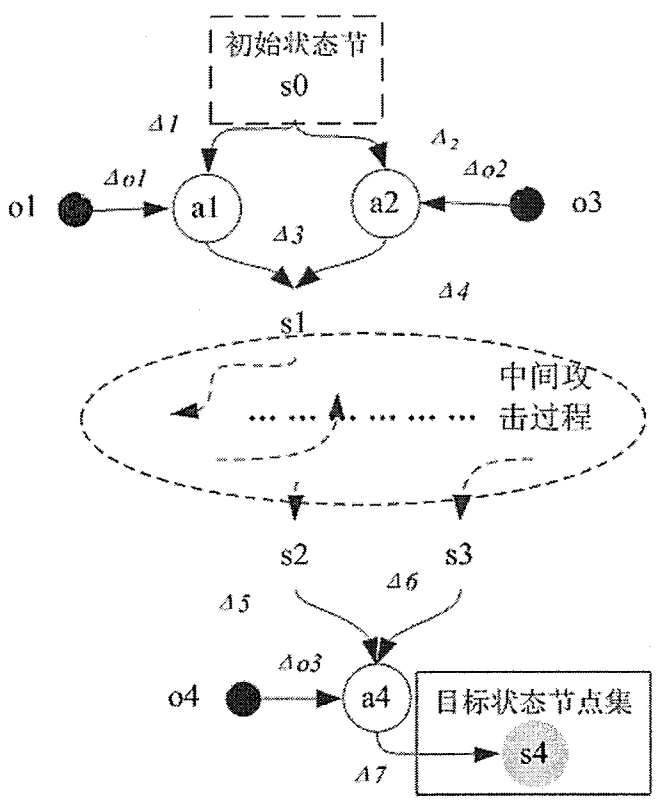


图 4.3: 攻击图示例

定义4.2. 边与节点的依赖关系：指向同一节点的边之间存在“与”和“或”的关系，具体定义如下：

- 依附于状态节点的边都是“或”关系。

- 依附于攻击动作节点的边有两种情况，如果，$e_i, e_j \in E_s$即从攻击状态节点到攻击动作节点，则e_i, e_j之间是“与”关系，表示要发动某次攻击必须同时满足所有前提状态。如果某条边$e_i \in E_o$，则该有向边与其他依附于该攻击节点的所有其他边是“或”的关系。

如图4.3中的状态节点s_2，表示该攻击状态能通过攻击a_2达到，也能通过实施攻击a_3达到。依附于a_4的边有三条，从s_2和s_3到a_4的两条边是“与”的关系，而从o_4到a_4的边与其他两条边是“或”的关系。本模型把观察事件节点到攻击节点之间的边认为是和其余所有依附于该攻击节点的边用“或”连接起来，这在现实中具有合理的解释：当人们观察到事件发生时，可以不用考虑其前提条件是否满足即可推断该攻击行为可能确实发生了；而如果没有观察到事件(比如IDS 被攻击者成功欺骗)，则可以通过其已经满足的前提条件来推断该攻击发生的可能性。

4.4　概率攻击图的构造

4.4.1　概率攻击图的依赖结构构造

概率攻击图的构造分为两部分：攻击图基本结构的确定和条件概率表CPT的生成。攻击图结构的生成目前大部分都依赖于专家知识库来生成，比如Ou等人[56]采用Horn逻辑描述攻击模板，可以利用漏洞和攻击之间的逻辑关系半自动辅助安全专家来生成大规模的攻击图结构。攻击图结构的生成是一个复杂的问题，本文的攻击图结构假定利用专家知识库来生成。专家知识库由网络拓扑和漏洞知识库共同组成。网络拓扑描述了一个网络信息系统中各个实体之间的网络连通性，如下表所示：

4.4.2　概率攻击图的概率转移表构造

上一节所描述概率攻击图中，条件概率表CPT是其中最重要的部分。张少俊[59]等人在其模型上也引入攻击行为发生的概率，攻击行为成功的概率以及攻击证据被观察到的概率$P(o_i|a_i)$，与本文介绍的概率攻击图最为相似。但是他们在计算过程中，假设攻击行为发生概率和成功的概率为0.5，$P(o_i|a_i)$为1.0。他们没有给出这三种概率在实际系统中的含义，且将$P(o_i|a_i)$定义为1.0 本质上与没有引入观测事件节点其结果是相同的。本文根据已有的静态漏洞分析知识库和安全监控系统动态生成的事件置信度作为确定条件概率表的依据，很好的与实际情况结合，使得后续的攻击意图推断算法更具有说服力。

确定Δ_s：表示攻击发生的概率。$\delta_{s_{ij}}$为在状态s_i下，攻击a_j发生的概率$P(a_j|s_i)$。

如果攻击方法是利用某个软件固有的漏洞，那么攻击发生的概率与该漏洞被利用的难易程度非常相关。一般来说，攻击方法越简单，其发生概率越大。NVD(National Vulnerability Database)发布了大量软件漏洞，通用安全脆弱点评估

表 4.3: 网络连通性示例表

From	To	Protocol/Port
0.0.0.0	Web Server	HTTP/80
	Email Server	SMTP/25 POP3/910
	GateWay	SSH/22
GateWay	All Machines in Trusted Zone	Base Network Protocol
Local User and Admin	GateWay	Base Network Protocol
	DB Server	SQL/1433
	File Server	NFS/111
	Web Server	HTTP/80
	Email Server	SMTP/25 POP3/910
	DNS Server	DNS/53
Web Server	DB Server	SQL/1433

系统CVSS(Common Vulnerability Scoring System)是一个能量化各个软件漏洞属性的工具。其中存取复杂度AC(Access Complexity)描述了发动漏洞攻击的难度，可以取值高，中，低。因此CVSS系统对漏洞的评估可以被作为确定$\delta_{s_{ij}}$ 的来源。

在内部攻击中，也存在大量攻击方法不是利用软件漏洞来发动，比如猜口令攻击、KeyLogger攻击等。这种攻击手段发生的概率可以通过安全专家指定，比如猜口令攻击其很容易实现，但不易成功。本文通过给每类攻击指定一个类似于CVSS 系统的AC属性一样的参数，来给出相应的概率取值。另外，也可以通过网络安全系统给出的攻击日志数量来确定。表4.4 给出了本文所使用的标准，如果一个漏洞在CVSS系统中的AC属性为“High”，或者某类攻击被安全专家标记为“High”，则说明该攻击实施的难度很大，满足预设条件时，其可能发生的概率被设置为0.2。如果某类攻击在一周内安全监控系统报警次数小于5次，则说明该类攻击手段很少被使用，其可能发生的概率被设置为0.1。具体的Δ_s的取值依据表如表4.4所示。

确定Δ_o：表示观察到的事件确定攻击正在发生的概率，即观测事件的置信度。δ_{o_i}表示攻击观察事件o_i 能证明a_i 正在发生的概率。

现代的信息安全监控系统能根据特定的规则检测某些具体攻击，也能根据用户的日常行为模式检测异常。前者通常预示着某些确定性的攻击正在发生，而后者给出的

表 4.4: 不确定性P1的取值依据表

AC属性（发生次数）	概率取值
High	0.2
Medium	0.6
Low	0.8
近一周<5次	0.1
近一周≤20次	0.5
近一周>20次	0.8

危险报警常常具有比较高的FP和FN，信息安全监控系统也常常能给出该类事件的置信度p。对于确定性的攻击o_i，δ_{o_i}可以取值为1；对有不确定性的异常事件o_j，δ_{o_i}可以取值为其相对应的置信度$p(o_j)$。δ_{o_i} 在概率攻击图中是一个延迟变量，其值在安全监测系统观测到具体事件时才能被决定。具体的取值见表4.5。

表 4.5: 不确定性P2的取值依据表

检测算法类型	概率取值
模式匹配	1.0
误用检测	$p(o_j)$延迟决定

确定Δ_a：表示攻击成功的概率。$\delta_{a_{ij}}$为攻击a_i成功后达到状态s_j的概率$P(s_j|a_i)$。

实际上攻击发生的概率和攻击成功的概率是相关的，比如攻击者选择一个软件漏洞进行攻击与其是否成功都是跟利用该软件漏洞的难易程度相关。但是它们又不是完全一样的，比如猜口令攻击，在信息网络中很容易发生这种攻击，但是成功的可能性却比较低。因此，这里将软件漏洞攻击的成功概率置为1.0，攻击发生概率中已经考虑了其难度，而对其他异常攻击，则根据专家知识库来设置。实际的取值见表4.6。

4.4.3 概率攻击图构造实例

图4.4中展示了两种内部攻击过程的实例。攻击场景A描述了内部漏洞利用攻击的过程图，恶意用户Malicious User (M)利用Web 服务器上的漏洞CVE-2002-0640 (URL缓冲区溢出)获取Web服务器的管理员权限，进而通过网络文件系统接口在文件服务器上写入木马程序，最后该木马被管理员Admin (A)不小心执行从而让恶意用户M可以窃取到管理员A计算机内的机密数据。该场景中的大部分动作都可以被安全监控系统检

表 4.6: 不确定性P3的取值依据表

攻击	概率取值
Vulnerability Using	1.0
Easy	0.9
Normal	0.5
Difficult	0.1

测到，M的攻击行为明显具备多步攻击的特性，他通过5个攻击步骤完成对机密信息的窃取企图，每一步攻击都具有明确目的，前后两次攻击动作是一种依赖关系。图4.4的攻击场景B描述了一个恶意的内部人员M利用职便，通过“猜密码”的方式进入了管理员A的个人电脑，并安装了一种具备Key Logger功能的恶意软件。A负责管理企业中的文件服务器(File Server)和其他资源。由此，M 获得了A在文件服务器上的用户权限，M通过A的权限登入了文件服务器，并下载机密文档。该场景下，除了恶意软件Key Logger行为可能会被杀毒软件检测到外，大部分的攻击行为都是在合法权限下进行的常规操作，难以被普通安全防护系统检测到。M也可能通过其他手段替代Key Logger获取文件服务器的密码，比如拷贝走A的密码维护文件。攻击场景B中，M的每一步攻击行为都具有伪装性，只能利用异常检测算法检测每一个步骤的行为异常。不少异常检测算法可以为每一个检测结果给出分值，指示算法对结果的置信程度，我们称为检测结果(观测事件)的置信度。置信度反应从一个观测事件推导出某步攻击真实发生的概率。传统的攻击图模型将检测结果作为确定性事件处理，无法在检测结果存在不确定性的情况下有效地推断攻击者的真实意图。

图4.4所示的网络拓扑中，网络被分为三部分，外网区域，内部办公区和内部服务器区。外网与内网通过防火墙系统逻辑隔离开来；内部服务器区部署一台网站服务器(apache2.2)，文件服务器(Linux2.4)和其他服务器；内部办公区部署一些终端和个人PC，包括一台模拟的恶意内部用户终端和管理员终端。内部服务器区上存在的漏洞包括网站服务器上apache的CVE-2006-3747(Apache mod_rewrite模块单字节缓冲区溢出漏洞)和文件服务器上存在CVE-2003-0252(Linux nfs-utils xlog()远程缓冲区单字节溢出漏洞)，两种漏洞的攻击复杂度分别为High和Low。实验环境中部署了一些安全监控软件，包括：Snort用来捕获各种漏洞利用攻击行为和网络行为异常事件，Tripwire部署在文件服务器上，用来监控文件系统的完整性，各个终端和个人计算机上部署OSSEC，用来监控PC端的木马活动和异常行为。

图4.5表示了图4.4两种攻击过程的完整攻击图。为了避免繁琐，图4.5中省去了条件概率表，除观测事件到攻击节点的条件概率在实际计算过程外，其他条件概率取值根

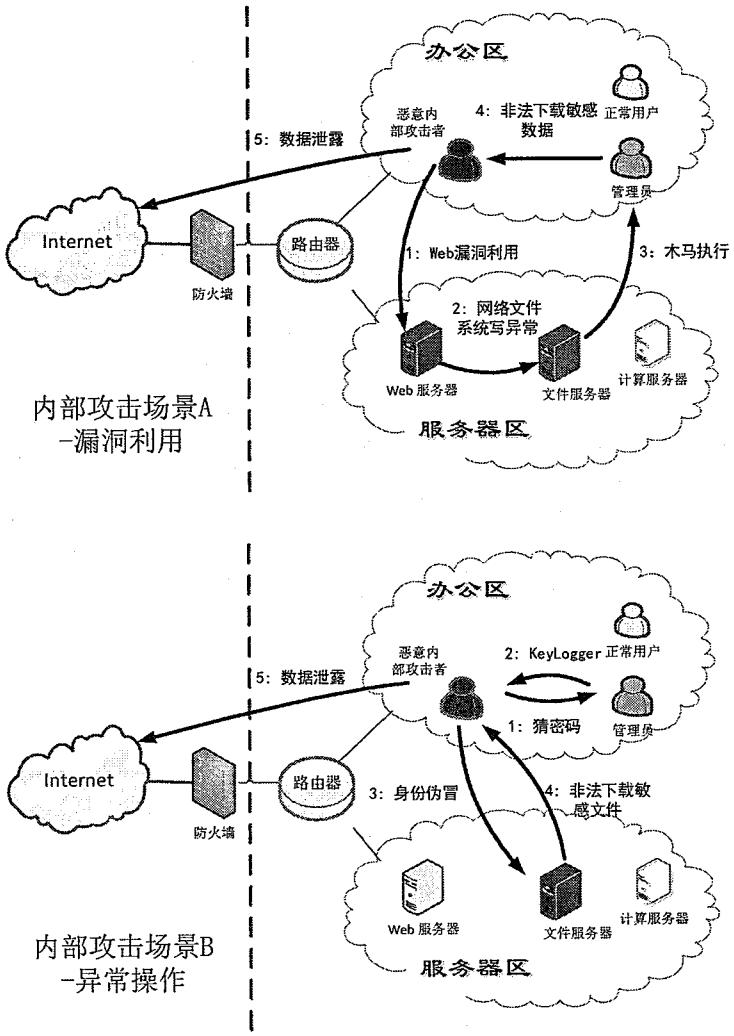


图 4.4: 内部攻击场景示例1

据上一节的定义方式在表4.7中给出。

4.5　概率攻击图上的概率推导

在概率攻击图定义的基础上，根据观测事件序列的进行概率推导首先需要为每个节点定义一个概率属性p。如果不考虑概率攻击图中的观测事件节点O及相关的边和概率E_o, Δ_o，该概率属性p代表各个节点发生的先验概率；当将观测事件节点O和对应的置信度Δ_o引入计算时，该概率属性p代表各个节点发生的后验概率。如Ou等人在文献[56]中讨论的，一般而言，网络威胁的攻击过程具有单调性(monotonicity)特征，单调性基于如下假设：攻击者不会轻易放弃或者丢掉已经获取到的权限。对应到本章讨论的概率攻击图而言，当看到越来越多的观测事件，节点被攻击的概率会越来越大。

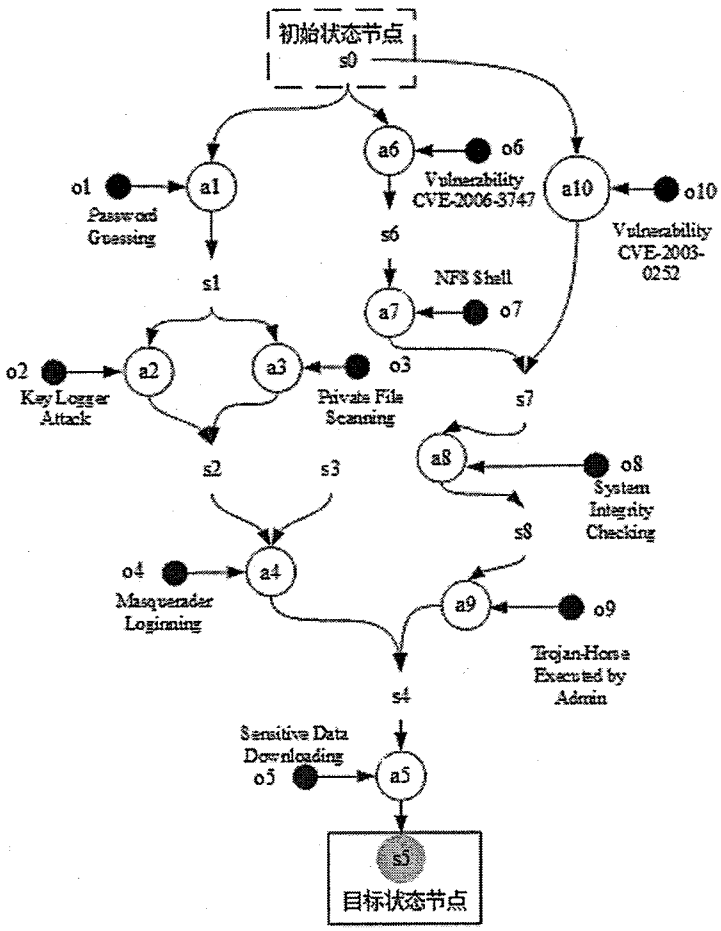


图 4.5: 对应于攻击场景1的概率攻击图

为此，我们也将节点的概率属性称之为节点的累积概率CP(Cumulative Probability)。

4.5.1 累积概率的定义

对概率攻击图中节点的累积概率具体定义如下：

定义4.3. 给定概率攻击图$PAG = (N, E, \Delta)$，节点集合N中各类节点的累积概率*CP(Cumulative Probability)* 被定义为：

- 观测节点的累积概率：

$$\begin{cases} CP(o_i) = 0, & if o_i = 0; \\ CP(o_i) = 1, & if o_i = 1. \end{cases}$$

即如果从安全监控系统观察到事件o_i，则该事件就是确定性发生了，累积概率为*1*；如果从安全监控系统中未观测到事件o_i，则该事件的累积概率为*0*；

表 4.7: 实验中的完整概率攻击图

边	概率取值	边	概率取值
(s_0, a_1)	0.9	(s_0, a_6)	0.2
(s_0, a_{10})	0.8	(a_1, s_1)	0.1
(a_6, s_6)	1.0	(a_{10}, s_7)	1.0
(s_1, a_2)	0.3	(s_1, a_3)	0.8
(s_6, a_7)	0.8	(a_2, s_2)	1.0
(a_3, s_2)	1.0	(a_7, s_7)	1.0
(s_2, a_4)	0.6	(s_3, a_4)	0.8
(s_7, a_8)	0.8	(a_8, s_8)	1.0
(a_4, s_4)	1.0	(s_8, a_9)	0.5
(a_9, s_4)	1.0	(s_4, a_5)	0.9
(a_5, s_5)	1.0	#	#

- 状态节点的累积概率：

$$\begin{cases} CP(s_0) = 1; \\ CP(s_i) = \oplus(pre(s_i)), \quad i \neq 0. \end{cases}$$

即初始节点的累积概率永远为*1*；其他状态节点的累积概率为其所有条件节点与边的之间的“概率或”操作结果。

- 攻击节点的累积概率：

$$\begin{cases} CP(a_i) = U(pre(a_i)), & if\ o_i = 0; \\ CP(a_i) = \oplus(U(Pre(a_i)),\ o_i), & if\ o_i = 1. \end{cases}$$

如果攻击节点a_i对应的观测事件节点没有看到，即$o_i = 0$，则$a_i = 1$的概率为攻击节点a_i所有的“与”操作结果；否则，攻击节点$a_i = 1$概率为a_i 所有的前提节点的“与”操作结果再与对应的观测事件节点$o_i = 1$的概率执行“或”操作。
上述定义中：

  - $\oplus(pre(s_i))$表示状态节点s_i的所有攻击条件节点之间执行“概率或”操作。假设指向状态节点s_i的所有边为$a_j - > s_i$, $a_k - > s_i$，即$pre(s_i) = a_j, a_k$，

$\oplus(pre(s_i))$的计算公式如*4.1:*

$$\oplus(pre(s_i)) = CP(a_j) \times \delta_{a_{ji}} + CP(a_k)\delta_{a_{ki}} - (CP(a_j) \times \delta_{a_{ji}}) \times (CP(a_k) \times \delta_{a_{ki}}). \tag{4.1}$$

- $U(pre(a_i))$表示攻击节点a_i的观测事件节点没有被观测到，即$o_i = 0$，所有条件状态节点之间执行“概率与”操作，假设指向攻击节点a_i 的所有边为$s_j- > a_i$, $s_k- > a_i$，即$pre(a_i) = s_j, s_k$, $o_i = 0$，$U(pre(a_i))$ 的计算公式如*4.2*所示：

$$U(pre(a_i)) = (CP(s_j) \times \delta_{s_{ji}}) \times (CP(s_k) \times \delta_{s_{ki}}). \tag{4.2}$$

- $\oplus(U(pre(a_i)), o_i)$表示攻击节点a_i的观测事件节点被观测到，即$o_i = 1$，的所有条件状态节点之间执行“概率与”操作，在与观测事件节点o_i的置信度执行“概率或”操作。假设指向攻击节点a_i的所有边为$s_j- > a_i$, $s_k- > a_i$，即$pre(a_i) = s_j, s_k$, $o_i = 1$，$\oplus(U(pre(a_i)), o_i)$的计算公式如*4.3*所示：

$$\oplus(U(pre(a_i)), o_i) = U(pre(a_i)) + CP(o_i)\delta_{o_i} - U(pre(a_i)) \times CP(o_i)\delta_{o_i}. \tag{4.3}$$

4.5.2 累积概率与攻击意图

攻击意图推断问题可以理解为计算攻击图中各个攻击目标节点的攻击概率，然后按照攻击概率对各个攻击目标节点进行排序，给出最大概率的攻击目标节点作为当前攻击过程中的攻击意图，次大概率的攻击目标节点当作候选列表提供给网络安全管理员。

因此，攻击意图推断问题的形式化定义如下：

问题4.1. 攻击节点对给定目标攻击节点s_{goal}的概率计算定义为给定概率攻击图$PAG = (N, E, \Delta)$，和观测事件序列$O = \{o_1, o_2, , o_n\}$ 的取值及其置信概率序列$\rho = \{\rho_1, \rho_2, ..., \rho_n\}$，其中$\rho_i$表示$o_i = 1$时的置信概率.
求$P(s_{goal}|PAG, O, \rho), s_{goal} \in G$.

根据累计概率的定义，有如下定理：

定理4.1. 给定概率攻击图$PAG = (N, E, \Delta)$，和观测事件序列$O = \{o_1, o_2, , o_n\}$的取值及其置信概率序列$\rho = \{\rho_1, \rho_2, ..., \rho_n\}$，其中$\rho_i$表示$o_i = 1$时的置信概率，$CP(s_{goal}), (s_{goal} \in G)$即为$P(s_{goal}|PAG, O, \rho)$.

Proof. 按照累积概率的计算方式，实际上每个状态节点和攻击节点的概率就是其累积概率，即$P(s_{goal}|PAG,O,\rho)$ 也就是$CP(s_{goal}), s_{goal} \in G$. □

由以上问题描述和定理，概率推断问题被归结为计算并寻找概率攻击图中最大攻击概率的目标节点集合问题。

4.6 内部威胁的意图推断

4.6.1 意图推断算法

在上述对节点累积概率的定义下，$CP(G)$即在观测事件序列下可能发生攻击意图的后验概率估计。因此，只需要计算每个攻击目标节点的累积概率，然后对其进行排序即可知道当前网络最可能发生的攻击意图。算法1在给定PAG结构和观测事件序列O以及对应的置信度序列ρ的情况下，计算每个节点的累积概率，并给出最大目标节点集合。

Algorithm 1 攻击意图推断算法Intention_Inferring

Require:

概率攻击图$PAG = (N, E, \Delta)$，观察事件序列$O = \{o_1, o_2, ,, o_n\}$的取值及其置信概率序列$\rho = \{\rho_1, \rho_2, ..., \rho_n\}$，其中$\rho_i$表示$o_i = 1$时的置信概率。

Ensure:

带cp属性值得PAG以及攻击概率最大的攻击节点。

```
function IntentionOfPAG(PAG, O, ρ)
    CalcProbability(PAG, O, ρ);
    intention_node.cp = 0;
    for each s ∈ S do
        if s ∈ Gands.cp > intention_node.cp then
            intention_node = s;
        end if
    end for
    return intention_node;

end function
//其中函数CalcProbability计算每一个节点的累积概率cp，具体定义如下：

function CalcProbability(PAG, O, ρ)
```

```
    Queue_Init(Q)   // 初始化队列Q
    Queue_Push(Q, s_0);   // 将初始节点压入队列中
    while Queue_Empty(Q) do
        n = Queue_Pop(Q);   //
        for each c_i ∈ ChildSetofn do
            Queue_Push(Q, c_i)
        end for
        if n ∈ S then
            n.cp = P_OR(n, pre(n))   //如果n在状态节点集中
        end if
        if n ∈ A then
            (o, p) = 从输入(O, ρ)中找到与节点n相关的事件和相应的置信概率;
            n.cp = ∏_{r_i∈Pre(n)} r_i.cp   //计算节点n的累积概率
            if (o, p) ≠ NULL then
                n.cp = n.cp + 1 * p − n.cp * 1 * p   //这里1是观察到的事件o的累积概率
            end if
        end if
    end while
    return 更新了节点的累积概率属性的攻击图PAG;

end function
```

在算法1中，概率“或”操作的计算如算法2所示。

Algorithm 2 概率“或”计算方法P_OR

Require:

概率攻击图$PAG = (N, E, \Delta)$，节点s和其前提节点集合$pre(s)$

Ensure:

输出节点s的累积概率CP的值

```
function P_OR(s, pre(s))
    for each c_i ∈ pre(s) do
        NOT_N = NOT_N × (1 − c_i.cp × δ)   //这里δ是节点c_i到节点s 的转移概率
    end for
    return 1 − NOT_N;
end function
```

4.6.2 意图推断增量算法

攻击意图推断算法-Intention_Inferring能够在任意观测序列下重新计算所有节点的累积概率。前面提到，网络攻击具有增量性特征。这意味着需要根据新的观测事件来实时更新攻击图中节点的累积概率，而不需要全部重新计算，以提高算法的效率。

新发现的观测事件会将某个观测事件节点o_i置为1，这将影响到对应的攻击节点的累积概率计算。该攻击节点又会影响到所有以其为前提条件的节点的累积概率计算，依次类推，最终重新影响攻击目标节点的累积概率计算。基于以上观察，意图推断的增量算法设计如3 所示：

Algorithm 3 攻击意图推断增量算法Intention_Inferring_Increment

Require:

1: 概率攻击图$PAG = (N, E, \Delta)$，新增加的观察事件序列$O' = \{o'_1, o'_2, ,, o'_n\}$的取值及其置信概率序列$\rho' = \{\rho'_1, \rho'_2, ..., \rho'_n\}$，其中$\rho'_i$表示$o'_i = 1$时的置信概率。

Ensure:

2: 带cp属性值得PAG

3: **function** CALCPROBABILITYIMC(PAG, O', ρ')

4: **for** each $o'_i \in O'$ **do**

5: CalcProbabilityImc_One(PAG, o'_i, ρ'_i);

6: **end for**

7: **return** PAG;

8:

9: **end function**

10: //其中函数$CalcProbabilityImc_One$计算一个新增观测事件对概率攻击图PAG的影响，具体定义如下：

11:

12: **function** CALCPROBABILITYIMC_ONE(PAG, o', ρ')

13: 找到o'对应的攻击节点a_k

14: // 重新计算攻击节点a_k的CP

15: $a_k.cp = \prod_{r_i \in pre(a_k)}(r_i.cp \times \delta_{a_{ik}})$

16: $a_k.cp = a_k.cp + 1 * \rho' - a_k.cp * 1 * rho'$ //这里1是观察到的事件o的累积概率

17: //以a_k为初始节点，计算后续节点的CP

18: $Queue_Init(Q)$ //初始化队列Q

19: $Queue_Push(Q, a_k)$; //将a_k压入队列中

20: **while** $Queue_Empty(Q)$ **do**

21: $n = Queue_Pop(Q)$; //

22: **for** each $c_i \in ChildSetofn$ **do**
23: $Queue_Push(Q, c_i)$
24: **end for**
25: **if** $n \in S$ **then**
26: $n.cp = P\_OR(n, pre(n))$ //如果n在状态节点集中
27: **end if**
28: **if** $n \in A$ **then**
29: $oldCond = \prod_{r_i \in pre(n)} r_i.cp$; //老的条件节点乘积;
30: $newCond = \prod_{r_i \in pre(n)} r_i'.cp$; //新的条件节点乘积;
31: $oldn = n.cp$ // 老的节点cp
32: **if then**$oldCond = oldn$
33: $n.cp = newCond$; // 该攻击节点的观测节点$o = 0$
34: **else**
35: $n.cp = newCond + \frac{oldn - oldCond}{1 - oldCond} - (newCond \times \frac{oldn - oldCond}{1 - oldCond})$ //根据以前的计算结果重新计算得到
36: **end if**
37:
38: **end if**
39: **end while**
40: **return** 更新了节点的累积概率属性的攻击图PAG;
41:
42: **end function**

以上计算过程的关键点在于两个地方：如果深度依赖新观测事件o'的节点n为状态节点，此时只需要用已经更新了的条件节点的累积概率来计算，即$n.cp = P_OR(n, pre(n))$。如果深度依赖新观测事件o'的节点n 为攻击节点，则需要借助在更新之前的PAG中的值来推断攻击节点的观测事件节点o观测到的情况。推断的方法是：如果$oldCond = oldn$，则说明$o = 0$，直接用$n.cp = newCond$来计算新的累积概率即可，否则说明$o = 1$，此时需要还原o对应的的置信度ρ，根据以前定义的计算操作4.3，可知$\rho = \frac{oldn - oldCond}{1 - oldCond}$。因此利用该值来重新计算$n.cp = newCond + \frac{oldn - oldCond}{1 - oldCond} - (newCond \times \frac{oldn - oldCond}{1 - oldCond})$。

4.6.3 意图推断算法复杂度分析

算法Intention_Inferring的时间开销主要在函数$CalcProbability$内部。该函数从初始节点出发，将节点的所有子节点进入队列，依次计算队列中节点的CP值。由于所有节点的CP计算只与其条件节点相关，因此，该算法能保证每个节点在计算CP 值时，其父节点的CP值都已经被计算过。该算法与图上的广度搜索算法复杂度相同，

为$O(N_num + E_num)$，即节点数和有向边数规模的和。

算法Intention_Inferring_Increment在最坏情况下与算法Intention_Inferring相同，因此算法复杂度也为$O(N_num + E_num)$。

4.7　攻击场景重构

攻击场景的定义是指攻击者发起攻击的过程。在内部威胁的概率攻击图里，某一个事件序列被观测到，可能预示着多种攻击场景发生，每种攻击场景有一定的发生概率。在存在不确定性的情况下，攻击场景重构需要计算每种攻击场景的发生概率，按安全管理员的需求给出概率最高的几条攻击路径。

在概率攻击图的模型里，攻击路径的形式化定义如4.5所示：

定义4.4. 攻击路径

*(1)*攻击路径$path$是概率攻击图PAG上的一个裁减子集，$path = (s_0, ..., a_j, s_i, a_{j+1}, ..., s_{goal})$，其中：$(a_j, s_i) \in E_a$ 且$(s_i, a_{j+1}) \in E_s$，$s_{goal} \in G$；即$path$是从初始节点到攻击目标节点的一条完整的路径。

*(2)*对概率攻击图PAG，所有攻击路径的集合称为$PATH$;

定义4.5. 最大概率攻击路径

最大概率攻击路径指这样一条攻击路径$\hat{path} \in PATH$，使得：

$$\hat{path} = \arg \max_{path \in PATH} (CP(path|GAP, O, \rho)).$$
$$CP(path|GAP, O, \rho) = \prod(s_i.cp) \times (a_j.cp); (s_i, a_j \in path).$$

由以上定义，为了计算最大概率攻击路径，必须计算$CP(path|PAG, O, \rho)$，路径的累积概率被定义为路径中所经的状态节点和攻击节点的累积概率之积。

4.7.1　最大概率攻击路径算法

根据攻击路径和最大概率攻击路径的定义，最大概率攻击路径计算算法如算法MaxProb_Path 4 所示。算法的关键步骤是从最大累积概率的攻击目标节点回溯，寻找其前提条件节点中具有最大累积概率的那个，直到节点没有前提条件节点。

Algorithm 4 最大概率攻击路径计算MaxProb_Path

Require:

1: 概率攻击图$PAG = (N, E, \Delta)$，以及概率攻击图中各个节点的累积概率$CP = n_i.cp|n_i \in N$。

Ensure:

最大概率攻击路径$\hat{path}$

function MAXPROBABILITYPATH(PAG, CP)

    $path = \{NULL\}$;
    Queue_Init(Q);
    $\hat{(g)} = \text{argmax}_{g \in G}(g.cp)$;
    add $\hat{(g)}$ to $path$;
    Queue_Push(Q, g);
    while
        do$n$ = Queue_Pop(Q);
        if $n \in A$ **then**
            Queue_Push($pre(n)$)
            add $pre(n)$ into $path$
        end if
        if $n \in S$ **then**
            $\hat{c} = \arg \max_{c \in pre(n)} (c.cp \times \delta_c)$ // 计算节点n的父节点中累积概率最大的一个;
            Queue_Push(c); // 节点出队列
            add c into $path$; // 将节点加入路径中
        end if
    end while
    $\hat{path} = path$
    return $\hat{path}$;

end function

4.7.2 算法复杂度分析

最大概率攻击路径计算算法$MaxProb_Path$与攻击意图推断算法$Intention_Inferring$相似，复杂度也是$O(N_num + E_num)$，即节点数和有向边数规模的和。

4.8 意图推断与场景重构实验

本节以已经构成的概率攻击图4.5为例，通过捕获的一些模拟攻击数据，给出概率攻击图的计算过程以及和其他方法检测结果的比较合分析，其结果验证算法的可行性，以及在缩小攻击报警范围方面的有效性和灵活性。

4.8.1 内部攻击实例分析

图4.5和表4.7确定了一个实验环境下的完整概率攻击图。通过在该实验环境部署的Snort，Tripwire和OSSEC等安全监控软件，可以观测到观测事件以及其对应的置信概率，以此为输入，即可通过前两节定义的算法1和算法4计算当前的潜在攻击意图及其最大概率攻击路径。实验结果如表4.8所示。

表 4.8: 三种观测序列下的意图推断与路径计算

编号	观测事件序列			最大概率攻击路径			
O_1	$\{null\}$			$\{s_0, a_{10}, s_7, a_8, s_8, a_9, s_4, a_5, s_5\}$			
O_2	$\{o_6:1, o_7:0.85, o_9:0.9\}$			$\{s_0, a_6, s_6, a_7, s_7, a_8, s_8.a_9, s_4, a_5, s_5\}$			
O_3	$\{o_2:1, o_4:0.9, o_8:0.5\}$			$\{s_0, a_1, s_1, a_2, s_2, a_4, s_4, a_5, s_5\}$			
状态节点	O_1	O_2	O_3	攻击节点	O_1	O_2	O_3
s_0	100	100	100	a_1	90	90	90
s_1	9	9	9	a_2	3.6	3.6	100
s_2	10.5	10.5	100	a_3	7.2	7.2	7.2
s_3	100	100	100	a_4	5.05	5.05	94.8
s_4	36.6	94.3	96.9	a_5	32.9	84.9	87.2
s_6	20	100	20	a_6	20	100	20
s_7	83.2	99.4	83.2	a_7	16	97	16
s_8	66.5	79.5	66.5	a_8	66.5	79.5	83.2
s_5	32.9	84.9	87.2	a_9	33.2	94	41.6
#	#	#	#	a_{10}	80	80	80

在表4.8中，针对三种情况对攻击意图进行推断和攻击路径计算。为了不失一般性，假定第一个实验中没有观测到任何事件，这相当于传统的网络安全风险评估方法，根据已有的漏洞知识和专家知识库可以发现系统中最脆弱的攻击路径。图4.6的子图a直观地体现了攻击者最可能的攻击路径以及置信度评估。在没有看到任何报警事件的情况下，可能存在攻击成功的概率是32.9%，最可能采取的攻击路径是$\{s_0, a_{10}, s_7, a_8, s_8, a_9, s_4, a_5, s_5\}$。第二个实验假设系统观测到了$o_6$,$o_7$和$o_9$事件，其对应的置信度分别为1.0,0.85和0.9。在此情况下，推测对目标节点s_5攻击成功的概率为84.9%，最可能的攻击路径为$\{s_0, a_6, s_6, a_7, s_7, a_8, s_8, a_9, s_4, a_5, s_5\}$，如

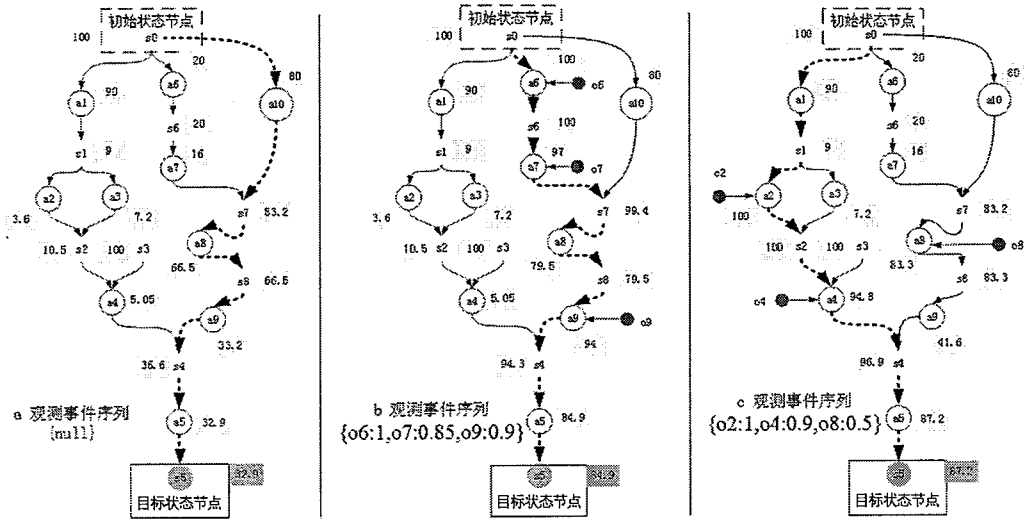


图 4.6: 意图推断与场景重构的过程

图4.6的子图b所示。第三个实验假设系统观测到事件o_2, o_4, o_8，对应置信度分别为1.0, 0.9, 0.5。在此情况下，推测攻击成功的概率为87.2%，最可能的攻击路径为$\{s_0, a_1, s_1, a_2, s_2, a_4, s_4, a_5, s_5\}$，如图4.6 的子图c 所示。需要说明的一点是，在第三个实验中s_1 的累积概率非常小，只有9%，但该路径上的其他累积概率都非常大，这可解释为安全监测系统极有可能没有捕获到“猜密码成功”事件，也说明本文所提方法能够处理在某些观测事件缺失的情况下推断攻击意图和攻击路径。

4.8.2 实验对比分析

贝叶斯网络算法被广泛应用在攻击图中检测攻击。叶等人[60]的工作在攻击图上计算累积概率，但他们的模型中没有考虑观察事件推导到攻击生的不确定性，观测事件与攻击发生是一一对应的，对应到我们的模型中即取值全为1。Xie等人[58]的工作中观测节点被有选择地引入，某些重要的攻击节点引入一个特殊的前提节点AAN。攻击节点的概率计算方式与本文所述方法不同，只有当AAN节点为真(被观测到)才用其前提节点进行与或计算得到攻击节点的概率。否则，其攻击节点的概率被置为0。本节将算法Intention_Inferring与叶和谢等人的算法进行对比。

从实验环境中安全监控软件的日志数据中，提取了100个观测序列及其相关的转移概率，加上4.3.1节的两次攻击序列O2 和O3，共有102个观测序列。本文实现了叶等人的算法(Inferring_Ye)和Xie等人的算法(Inferring_Xie)，其中Xie的算法分别选择a7(1_AAN)，a7和a9(2_AAN)作为具有AAN节点的攻击节点。将这102个观测序列作为输入，在为目标节点选择不同的置信度阈值情况下，三个算法的检测结果情况如下表4.9。

从结果对比表4.9中，首先观察到的是我们的算法(Inferring_Our)的报警数量明显要比其他两种算法少。通过分析那些在其他算法中报警但我们的算法中没有被检测出来的案例，发现很多在叶算法中被检测命中观测序列，其观测事件的置信概率都很低，如第10、30和31个观测序列($O_{10} = o_2 : 0.1$, $O_{30} = o_4 : 0.1$, $O_{31} = o_4 : 0.2$)，在Inferring_Ye和Inferring_Xie 1AAN 都检测命中，而在Intention_Inferring 中检测结果的置信度分别为35.5%, 38.7%和44.4%。这种差别的原因在于其他两种算法只考虑观测事件是否发生，而没有考虑观测事件的置信度，这也正是内部攻击攻击行为检测会造成大量报警日志的原因。通过利用安全监控软件中异常检测算法提供的观测事件的置信概率，算法Intention_Inferring能更有效地检测攻击威胁。

另一方面，如果网络安全管理员也可以通过调节攻击目标节点的置信阈值实现捕获更多攻击。上一小节的实验已经表明在没有任何输入事件($O_1 = null$)的情况下，概率攻击图的目标节点受到攻击的概率是32.9%。只要选择的置信阈值等于或低于32.9%，所有的攻击行为都会被检测到。

其次，算法Intention_Inferring对阈值变化的敏感性要优于其他两种算法。当调节阈值从0.5到0.9时，算法Intention_Inferring给出的检测结果数量从38降到2，且每一次阈值调节带来的变化都很明显。而算法Inferring_Ye和Inferring_Xie 则表现得不敏感。这也是因为这两种算法仅考虑了攻击图结构的不确定性，这些不确定性都是在攻击图构造过程中就已经固定下来了，因此，以上两种算法的计算过程对输入观测序列的敏感性不高。而算法Intention_Inferring更重要的是将运行期监控到的观测事件及其置信概率作为输入，可以实时计算攻击意图和攻击路径。算法对阈值选择的敏感性可以帮助网络安全管理员更快地找到嫌疑最大的攻击行为，提供了良好的可配置性。

表 4.9: 意图推断结果对比表

置信阈值	Intention_Inferring	Inferring_Ye	Inferring_Xie	
			1_AAN	2_AAN
0.5	**38**	64	64	34
0.6	**26**	44	34	34
0.7	**22**	34	34	34
0.8	**17**	34	34	34
0.9	**2**	10	10	10

4.9 本章小结

内部攻击行为的检测面临着复杂性和不确定性的挑战。本章首先讨论了已有攻击

图在描述内部威胁攻击过程中的不足，然后详细分析了内部威胁中存在的三种不确定性，设计了概率攻击图模型，给出详细的形式化定义。本章接着描述了如何在目标网络中根据安全专家知识库和各种安全监控软件的实时检测结果建立概率攻击图，推断攻击意图并计算可能的最大概率攻击路径。本章提出的模型和算法考虑了监测事件的不确定性，可以利用现有的安全监测系统的异常检测功能，应对内部攻击行为的伪装性和多步骤性带来的挑战。实验结果表明本文的工作能够有效推断攻击意图和计算攻击路径，减少不可信报警数量，为网络安全管理员提供良好的可配置性。

第五章　概率攻击图上的动态安全防护策略

5.1　研究背景与问题概述

网络安全风险评估在应对网络安全问题方面发挥着积极的作用，也是网络安全研究的热门领域之一。而最优安全防护策略研究是网络安全风险评估研究中非常重要的问题之一。网络安全风险评估分为静态风险评估和动态风险评估。静态风险评估指根据当前的网络状况，包括网络拓扑、漏洞存在情况、网络访问控制策略、网络服务依赖等情况，分析当前网络信息系统所面临的威胁风险。静态风险评估对应静态安全防护，主要解决的问题是如何在满足一定代价条件限制的情况下，采用最优的安全防护措施对网络信息系统的组成部分进行防护，保护企业、组织或者政企机构的核心资产。防护的目标也因为核心资产的类型不同而不同。如果核心资产为基础设施，那么防护的目标为防止恶意用户取得基础设施的控制权限；如果核心资产为企业提供的服务，那么防护的目标是阻止服务的可用性遭到破坏；如果核心资产为数据或者文件，那么防护的目标是保护数据的隐私性。动态安全风险评估指在静态风险评估所参考网络基本信息的基础上，如何根据实时观测到的告警事件对网络安全威胁的态势进行实时评估。动态风险评估对应的防护问题是动态安全防护，要求根据实时观测到的告警事件调整防护策略，使之能够最大限度降低核心资产受到攻击的风险。静态安全防护和动态安全防护研究的唯一不同之处在于后者考虑到网络信息系统当前的安全状况，能够动态调整安全策略，将网络风险减到最低。

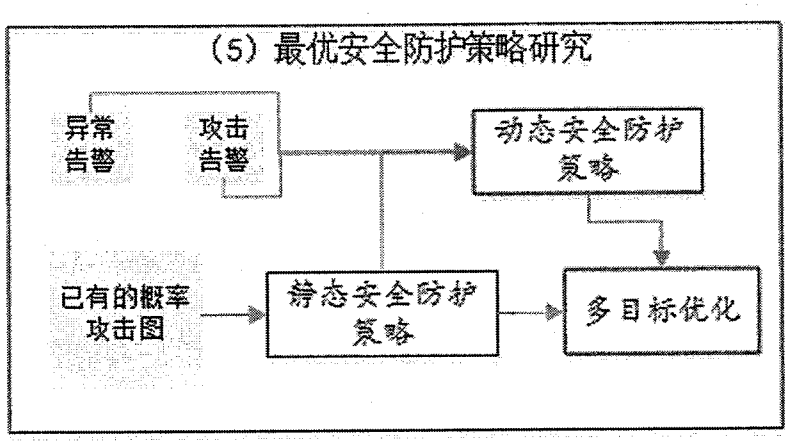


图 5.1: 安全防护策略研究框架

安全防护策略研究的框架关系如图5.1 所示。静态安全防护策略计算在确定了概率攻击图之后即可进行；动态安全防护策略计算在静态安全防护策略的基础上，能根据

观测到的攻击事件或者异常事件进行动态调整。无论是静态安全防护策略还是动态安全防护策略都需要满足多目标优化限制条件，如满足一定的条件代价，保证一定的服务可用性等等。

无论是静态安全防护策略研究还是动态安全防护策略研究，传统的方法主要集中在研究攻击图上的最优安全防护策略计算。Jha 等人[61] 最先在状态攻击图上计算最优安全防护策略，他们将关键集定义为这样一些原子攻击组成的集合：当移除了属于关键集中的原子攻击之后，攻击者不能从初始攻击状态节点到达攻击目标状态节点。而最小关键集即为针对某个攻击目标所有关键集中最小的集合。关键集和最小关键集的定义类似于连通图上的“割”的概念。在一个连通图$G = V, E$ 上，如果原始节点集为V，选择一些节点集S使得$s \in S$，$T = V - S$，则S 到T 的边为S 到T 的割，称为$[S, T]$。在攻击图中，可以将节点和边做一个映射即可将将关键集转化为割的概念。该文最后将问题转化为求一个能够覆盖最小关键集的安全防护策略集合的问题。

然而，由于攻击动作之间的有很强的依赖关系，最小关键集依然包含了大量无关和冗余的攻击节点。比如最小关键集中包括一个ftp 相关的漏洞攻击，而该ftp漏洞攻击依赖于具有ftp漏洞的软件安装，并且依赖于攻击者对该ftp 服务器的网络访问。这两个依赖条件，特别是第二条很可能成为最小关键集中另一个节点的依赖条件。在复杂的攻击图中，要修复最小关键集中的攻击，可能需要做大量重复的操作。基于此种观测，Wang[57] 和Noel[62] 等人研究了攻击之间的依赖关系，并将每一个攻击当作一个贝努利变量，只能取值0,1，表示攻击是否发生。通过攻击之间的依赖推导，每一个攻击目标可以用一些基本攻击的谓词合取范式来表示。基本攻击指不依赖与任何其他攻击的攻击节点，也称为初始攻击节点。初始攻击节点无法应付动态可变的情况，如果某一步初始攻击未被检测到，或者被绕过（这种情况在内部威胁场景下尤为常见），那么基于初始攻击节点的防护策略计算方法就会失效。

另外，以上的两种方法只考虑了网络配置条件，完全忽略了安全控制手段的代价和收益问题。Poolsappasit和Dewri[64] 等人将网络安全防护的计算扩展到一种多目标的优化问题。他们试图在安全控制代价和整体收益方面寻找一个平衡点，通过一个贝叶斯网络模型来描述攻击过程，并在此基础上进一步提出使用一种基因算法计算多目标优化问题，取得不错的效果。程叶霞等人[90] 也基于攻击图，提出了攻击可能性、攻击实现度、脆弱性程度和脆弱点关键度指标等四个网络安全评估的指标，因此，网络安全加固就转化为一个面向网络安全的多目标优化问题。

以上算法在在一定程度上被证实是有效的，但是不太适合内部威胁的情况。首先，内部威胁的首要特征是在内部发起，在描述内部攻击过程的攻击图上，很难找到初始节点的概念，因为攻击可能从任意的地方发起。比如一个传统的DB 服务器要抵御攻击，可能限制用户的访问权限，但是内部攻击者可以利用职便，窃取管理员的账户信息进行访问。因此，面向内部威胁的攻击图仅仅从源头上加强防护是远远不够的。其

次，内部威胁行为由于具有很强的伪装性，导致对攻击的检测结果具有不确定性，甚至缺失对攻击的观察（数据缺失），Poolsappasit 和Dewri[64] 和程叶霞等人[90] 的攻击图模型没有足够模拟内部威胁面临的不确定性。而且程叶霞等人提出的多目标优化指标彼此之间并不是正交的，存在一定的相关性。因此，这些模型不太适合对内部威胁进行最优安全防护策略的动态计算。

本章在前面对概率攻击图模型定义的基础上，将其扩展为方便描述安全防护策略的$MPAG$。然后在该模型基础上讨论了安全防护策略问题，研究在概率攻击图上的多目标优化问题，包括威胁风险、安全防护措施的有效性，代价等，也讨论了静态安全防护策略计算和动态安全防护策略计算，并提出一个贪心算法统一解决安全防护策略问题，最后通过实验验证算法的有效性。

5.2　安全防护策略研究问题

5.2.1　多目标优化

在攻击图上的安全防护策略计算需要考虑多个方面的权衡。程叶霞等人[90] 在攻击图上提出了四个评价指标，攻击实现度指不同弱点对应的成功实现程度；攻击可能性指攻击者从攻击起点状态s_0 出发选择到达攻击目标状态集合I_S的概率；攻击图的脆弱性程度指标是指实施攻击行动的难易程度；击图的脆弱点关键度指标是指脆弱点存在与否对整个网络环境的影响程度。最后程叶霞等人将安全防护问题表述为在满足一定的限制条件下针对该四个指标的最大最小化问题。

Dewri 等人[65] 考虑两个优化目标：第一个目标是风险影响(Residual Damage)，计算方式定义为$RD(\vec{T} = V_{root})$，表示当前防护策略$\vec{T}$ 条件下的以$root$为根的攻击树的威胁风险值。第二个目标为总共的安全控制代价(Total Security Control Cost)，计算方式定义为$SCC(\vec{T}) = \sum_{(i=1)}^{m}(T_iC_i)$，总共的安全控制代价是每个安全措施$T_i$ 的代价之和。

从实际方面考虑，安全防护策略的选择总是受到几个方面的限制。安全防护措施是一套安全策略或者工具，可以避免某些攻击发生或者降低攻击成功的概率。比如为了抵御“猜口令攻击”，网络安全管理人员制定了“口令长度及修改周期”的策略，要求所有的员工每两周修改一次密码，密码的长度不少于8 个字符。该策略能够有效降低“猜口令攻击”的成功概率，密码长度让“猜口令攻击”的成功概率大大降低。另外一个例子在内网中经常被使用的策略是设置屏保的时间不超过10分钟，即当系统在10 分钟没有任何交互操作时，系统自动进入屏保状态，用户再进入需要重新输入密码。这种防护措施能抵御“身份冒用攻击”，但效率跟屏保的时间有关系，如果时间太长，身份冒用者很可能早已完成了攻击。如果时间太短，可能会影响用户的使用，如果设置2分钟的话，用户还没浏览完一个页面，系统就自动进入了屏保状态，这将是一件非常令人讨厌的事情。

由以上的讨论，初步可以确定安全防护措施有三个性质：1)可避免攻击的发生，或

者降低攻击发生的可能性；2)安全防护措施有一定实施成本，或者消耗人力或者管理成本，或者要求有一定的资金投入；3)安全防护措施可能影响系统的使用性，或者成为影响系统或者服务的可用性。关于影响服务可用性方面一个最极端的例子是：一个企业的web 应用存在已知的漏洞，但该漏洞的修复需要一定时间。当前一个可选择的安全防护措施是切断web 应用的网络连接，该防护措施能够完全抵御外部攻击，但是却以丧失整个web 应用的可用性为代价。是否值得这样去做取决于管理者对漏洞的危害性和服务的重要性之间的利益权衡。

为了寻找一个统一简单的模型，将安全防护措施的实施成本和系统可用性的影响可以合起来，称为安全防护措施的**实施代价**。安全防护措施减低攻击风险的性质称为**实施效益**。

5.2.2 静态和动态安全防护策略计算

一套有效的安全防护策略可以以一种前摄性的方式或者以一种反应性的方式被实施。给定一个安全防护措施集合M，可以设计一套安全策略(Security Plan)能够最大限度减少攻击风险。这套安全策略能够在系统部署之前设计启用，也能在遭遇到攻击事件时设计启用。前者称为静态安全防护，后者称为动态安全防护。

- 静态安全防护策略计算对应于静态安全风险评估。为了做风险评估，首先要确定系统的特征，潜在的威胁源，攻击者的能力等情况。威胁源即系统中可能被攻击的点，每一次攻击有一定的前提才能开始实施，被称为攻击的资源要求，实施成功后会获取更多的资源。攻击是否能发生还依赖与攻击者的能力，通常这种判断都是基于网络安全管理员对威胁源能否被利用做攻击的主观推断。静态安全风险评估是计算系统受到攻击的先验概率，寻找一套安全策略M 使得系统受到攻击的先验概率最小的计算，被称为静态安全防护策略计算。

- 动态安全防护策略计算对应于动态安全风险评估。已部署的系统会在其生命周期内直接经历的攻击事件，利用攻击图来处理告警事件关联、处理告警事件缺失以及预测未来的攻击是动态安全风险评估的研究范围。当一个攻击发生时，安全管理员希望调查该事件对整个网络系统的风险影响是什么，有多大。贝叶斯推理通常被用来更新攻击图的风险概率。动态安全风险评估是在观测到告警事件时，计算系统受到攻击的后验概率，寻找一套安全策略M使得在观测到告警事件时系统受到攻击的后验概率最小的计算，被称为动态安全防护策略计算。

5.3 最优安全防护策略计算

本节讨论如何实时计算最优安全防护策略问题。首先讨论了在概率攻击图上的安全防护策略属性扩展模型MPAG(Measure Probability Attack Graph)，然后讨论了在MPAG基础上进行最优安全防护策略算法设计和算法复杂度的分析。

5.3.1　安全防护策略概率攻击图

第四章定义的概率攻击图PAG用攻击节点、状态节点和观测节点以及节点之间的边与概率计算关系刻画了内部威胁中存在的三种不确定性，并定义了节点累积概率的概念及其概率推导方法。

为了适用于计算安全防护策略问题，本节将原来定义的概率攻击图做了扩展，加入了安全防护策略节点。安全防护策略节点具有三个属性，分别用来表示是否启动某个安全防护策略，启动的代价以及预计的效果等。具体而言，扩展后的安全防护策略的形式化定义如下：

定义5.1. 安全防护策略概率攻击图被定义为一个有向无环图
$MPAG = (S, A, O, E, \Delta, \Gamma, M, \Phi, C)$，其中：

- S是状态节点的集合，即是PAG中的N的一个子集。每一个状态节点代表攻击者当前状态下所拥有的攻击资源和系统权限，同样也表示网络信息系统被攻击的程度。具体而言，$S = S_{in} \cup S_{goal} \cup S_{ex}$，其中$S_{in}$表示外部攻击者最初所处的状态集合，但在有内部攻击存在的条件下，初始状态节点的意义不大，因为攻击可能从任意情况下发起。为了与传统的攻击图定义保持一致的形式，我们保留了初始状态节点的定义。S_{goal}表示目标状态节点集合，与PAG中的定义相同。S_{ex}表示攻击者攻击过程中所达到的状态节点。所有$S_{goal} \cup S_{ex}$中的节点都以一个或者几个攻击节点为条件，称其为状态节点的条件节点。同时，所有$S_{in} \cup S_{ex}$中的节点都是攻击节点的攻击条件节点。同样，$pre(s)$来表示状态节点的攻击条件节点集合。

- A表示攻击节点集合。给定$s_{pr}, s_{po} \in S$，$a : s_{pr} \rightarrow s_{po}$被定义为一个原子攻击。其中$s_{pr}$被称为原子攻击$a$的条件状态节点，表示原子攻击$a$发生所需要的前提条件，$s_{po}$称为原子攻击$a$的后继状态节点，表示攻击成功后的状态。一次攻击在前提条件的spr下发生，成功后能进入状态节点。原子攻击可能以多个状态节点为状态条件节点，用$pre(a)$来表示攻击节点的状态条件节点集合。

- O代表观测事件节点集合。为了方便计算，定义集合A中的元素与集合O中的节点是一一对应的，表示攻击a_i发生时被安全监控软件可能观测到的事件。这与PAG的定义是相同的。

- E, Δ的定义与PAG完全相同，E代表着各个节点之间的边，Δ则蕴涵着三类边之间的不确定性概率。这与PAG的定义是相同的。

- Γ是一个元组集合$< S \cup A, d_i >$，表示状态节点或者攻击节点与其条件节点集之间的关系。关系d_i的取值范围为$< AND, OR >$。按照前面的定义，状态节点的

条件节点是攻击节点，通常认为，一个状态可以由多种攻击到达，因此约定状态节点s_i的所有条件节点$(a_i, a_j, ,)$之间的关系d_i取值为OR。而攻击节点的条件节点是状态节点，如果要实施一个攻击，必须得到全部前提条件才可以开始，即可设置攻击节点a_i的所有条件节点$(s_i, s_j, ,)$之间的关系d_i取值为AND。每一个攻击节点a_i都与一个观测事件节点o_i关联，定义o_i与a_i的状态条件节点$(s_i, s_j, ,)$的关系为OR。这样定义是合理的，因为不管是否条件节点是否满足，如果观测到攻击事件，则相应的攻击可能已经发生。$< AND, OR >$的取值决定了概率推导计算方法是不一样的，这在计算累积概率的定义中已经体现。

- $M, \Phi, C3$个属性是为了计算最优安全防护策略引入的。其中，M表示防护策略，不失一般性，假定针对一个攻击a_i都有一个相应的防护策略m_i来防止攻击。m_i是一个贝努利变量，取值空间为$0,1$，分别表示启用和不启用该防护措施。启动防护措施后对攻击图的影响由Φ表示，Φ中的元素与M的元素一一对应，某个元素ϕ_i的值表示对攻击a_i启用的防护策略m_i后影响攻击成功概率δ_{a_i}的程度，其取值范围为$[0,\ 1.0]$，启动防护措施m_i后，$\delta'_{a_i} = \delta_{a_i} \times \phi_i$，也就是说，$\phi_i$越小，其防护能力越强。对于某些利用漏洞的攻击来说，给漏洞打上补丁后，其攻击成功概率可以为0，即其可取值为0；而对于一些内部攻击手段，比如文档访问行为异常，能通过限制某些类型的下载请求来减轻该类攻击，其攻击成功概率可能下降一定的比例。该值一般由网络管理员依靠经验来配置。

- C表示安全防护措施的代价集合，c_i为安全措施m_i的实施代价，c_i的取值是一个实数。

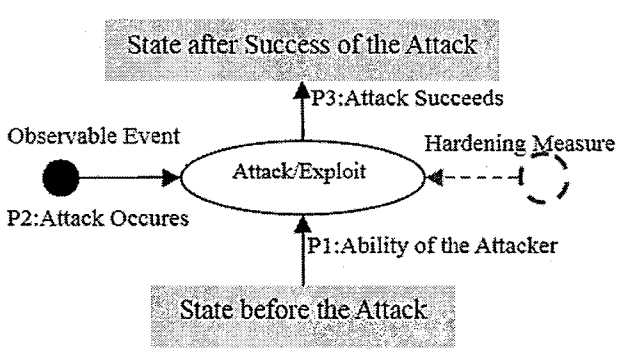


图 5.2: 安全防护策略图的影响

图5.2 展示了安全防护策略与三类不确定性之间的关系，与图4.2不同的地方为对每一个攻击节点增加了一个安全防护措施节点，红色的虚线圈表示，其属性用三元

组$[m_i, \phi_i, c_i]$来表示。每个攻击节点都对应一个这样的属性节点，只有当对应的m_i被设置为1时，该节点的相应值才会参与计算。

5.3.2 最优安全防护策略算法

在上述$MPAG$的顶一下，最优安全防护策略的计算问题被如下形式化的描述：

问题5.1. *最优安全防护策略问题定义为在给定1)安全策略概率攻击图$MPAG = (S, A, O, E, \Delta, \Gamma, M, \Omega, C)$; 2) 观测事件序列$O = \{(o_1 : p_1), ...(o_i : p_n), ...(o_n : p_n)\}$, o_i=1, 表示发现观测事件, p_i是相应的置信度; 3)预算限制C'下, 求:*
一个$M' = (m_1, ...m_k, ...m_n)$的取值, 满足:
(1)$P(s_{goal}|MPAG, O, M')$ 最小;
(2)$\sum_{i=1}^{n}(c_i m_i) \leq C'$.

在$MPAG$上定义的最优安全防护策略问题是一个NP-难问题，这一点可以通过将0-1 背包问题归约于最优安全防护策略问题来证明。

定理5.1. *如果T 代表问题5.1所定义的一个最优安全防护问题，R代表经典的0-1 背包问题，则$R <= T$.符号<=表示可归约，$R <= T$ 即表示问题R可归约为问题T。*

Proof. 1. 0-1 背包问题的定义是：有n 种物品，物品i的重量为w_i，价格为p_i，背包所能承受的最大重量为W。限定每种物品只能选择0 个或1 个，用贝努利变量x_i表示。求：
(1) $max(\prod_{i=1}^{n}(p_i \times x_i))$,
(2) 在限制条件$\prod_{i=1}^{n}(w_i \times x_i) \leq W$下。

2. 很容易将取某个物品x_i 可与采用安全措施m_i 相对应，其重量w_i 可与安全措施的代价c_i相对应。

3. 假定p_i等于给定的攻击图中，启动m_i 前后的攻击目标概率差值，即

$$p_i = P(s_{goal}|O, m_i = 0) - P(s_{goal}|O, m_i = 1),$$

则0-1 背包问题被归约为最优安全防护策略问题。因此T 问题是NP-难的。

□

需要说明的是，实际的最优安全策略问题比0-1 背包问题更难，主要原因在于启动m_i所引起的攻击目标概率差值并不是独立的，即有下列不等式：

$$P(s_{goal}|O, m_i = 0) - P(s_{goal}|O, m_i = 1) \neq P(s_{goal}|O, m_j = 1, m_i = 0) - P(s_{goal}|O, m_j = 1, m_i =$$

但这并不影响上述证明，它足以说明实际的最优安全策略问题是一个NP-难的问题。

最优安全防护策略的计算是一个非常困难的事情。如果用暴力算法，M的取值一共有2^n 种不同的取值情况，其复杂度将是$O(2^n \times (|S|+|A|+|E|))$，其中$O(|S|+|A|+|E|)$为在每一种取值情况下，攻击目标概率计算的算法复杂度。

在给定的概率攻击图上，通过对所有的安全防护策略进行遍历，计算$P(s_{goal}|O, m_i = 0) - P(s_{goal}|O, m_i = 1)$ 之间的概率差，选择使概率差最大的安全防护措施m_i。基于该贪心选择，可以设计出简单的贪心算法解决该问题。该算法的详细定义如算法5所示。

Algorithm 5 OMSN(optimal_measures_select_normal).

Require:

概率攻击图$MPAG = (S, A, O, E, \Delta, \Gamma, M, \Omega, C)$，观测事件序列$O = \{(o_1 : p_1), ...(o_i : p_n), ...(o_n : p_n)\}$，$o_i$=1，表示发现观测事件，$p_i$是相应的置信度; 预算限制$C$';

Ensure:

$M = m_1, m_2, , , m_n$，使得$P(s_{goal}|MPAG, O, M')$ 近似最小.

function OMSN($MPAG, O, C'$)

    $M' = 0,0,\cdots 0$; //初始化M'为全0;

    $left\_C = C'$;

    while ($left\_C$¿0) **do**

        $P\_SGoal = IntentInferring(MPAG, O, M')$; //重新计算在当前防护措施集合下的攻击目标概率，首次计算M'为全0 ;

        $max_margin = 0$; //启动m 后的最大概率差

        $m\_i = 0$; //保存具有最大概率差的m的索引

        for $i = 0 : n$ **do**

            if $m\_i == 1$ **then** //m_i已包括在M'中

                continue;

            end if

            $P\_SGoal\_m = IntentInferring(MPAG, O, M')$;

            if $c_i < left\_C\ AND\ (P\_SGoal - P\_SGoal\_m) > max_margin$ **then** //如果当前节点的代价不超过剩下的代价上限，且概率差比上一个m大，更新m的索引和最大概率差

                $m\_i = i$;

                $max_margin = P\_SGoal - P\_SGoal\_m$;

            end if

        end for

        if $m\_i == 0$ **then**

```
21:            break;
22:        else
23:            M'的第m_i个元素设置为1;
24:            left_C = left_Cc_i;   //更新剩余的代价
25:        end if
26:    end while
27:    return M';
28: end function
```

5.3.3 算法复杂度分析

$IntentInferring(MPAG, O, M')$采用一个深度优先算法，其复杂度是$O(|S|+|A|+|E|)$，即所有状态节点和攻击节点的总和加上边的数量。$OMSN(MPAG, O, C')$有两层循环，外层的FOR 循环次数是$n$，内层WHILE 循环将执行直到$left_C$不满足任何防护测量的条件停止，但最大循环次数不会超过n。此处n 表示攻击节点的数量，也就是是防护策略节点的数量。因此OMSN 的算法复杂度为$O(n^2 \times (|S|+|A|+|E|))$。

5.4 实验结果及分析

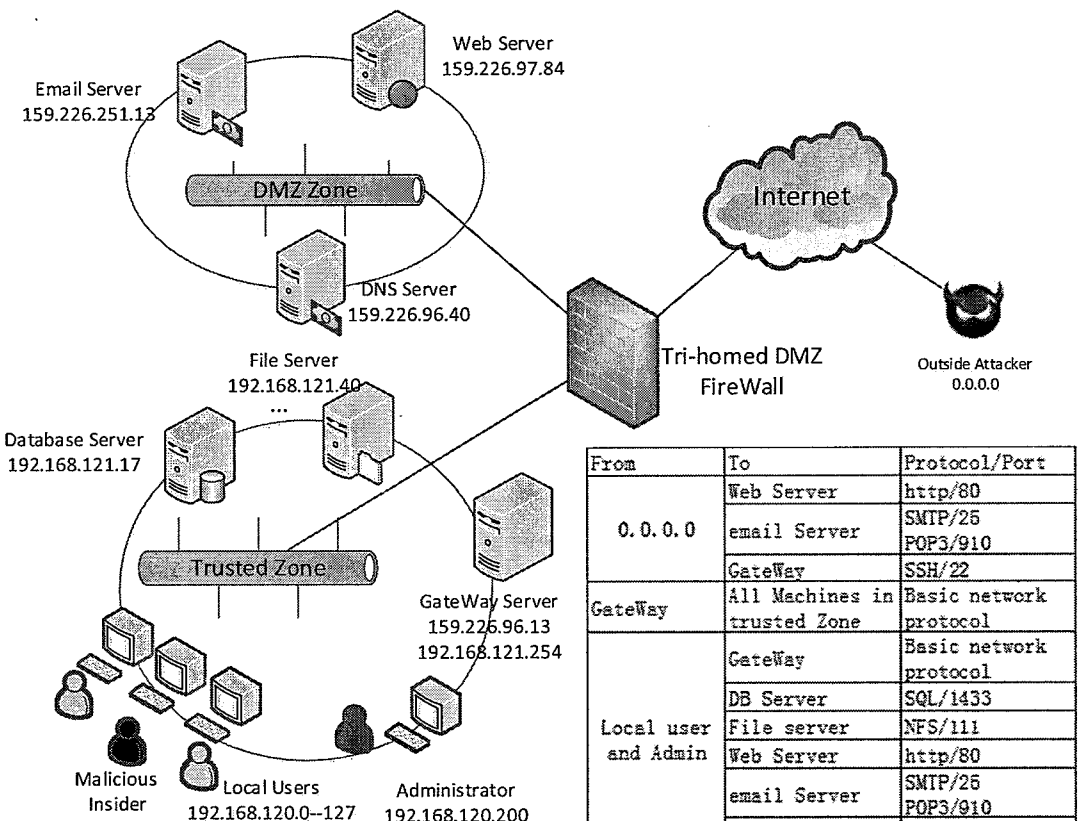
本节首先以图5.3 所代表的网络拓扑为例，介绍在实验环境中真实搭建的网络环境，以及在此网络环境中的安全漏洞和内部攻击手段。然后给出在此网络环境下的MPAG 模型，并根据经验和专家知识确定各种概率的取值。最后，对模拟的攻击事件以及对应的置信概率计算相应的安全防护策略集合，验证最优安全防护策略算法的有效性与实时性。

5.4.1 实验环境设置

在图5.3 所示的网络拓扑中，网络由三个部分组成，Internet 区域、DMZ 区域和内部受信网络区域。Internet 区域是完全不可信的网络，可能存在恶意的外部攻击者；受信网络区域是完全可信的内部网络区域，提供本地办公、数据库服务、文件集中服务器以及NAT 网关等功能；DMZ 区域处于不可信Internet 区域和完全可信的内部网络区域之间，企业一般将对外提供的网络服务部署在此，比如企业的Web 服务，邮件服务和DNS 服务，同时该区域与内部可信的网络区域的通信受到限制，只能提供有限的访问，比如Web 服务器可以通过SQL查询访问受信网络区域中的SQL Server 服务器。企业通过防火墙系统将各个区域逻辑隔离开来，并配置其间的网络通信规则，详细可见图5.3 中的表格。

根据漏洞扫描的结果和内部威胁的经验知识，系统可能存在漏洞或者异常攻击行为被列在表5.1中，既包括SQL 注入、IIS 漏洞，也包括身份冒用和SQL 数据访问异常

等内部威胁检测中经常讨论的威胁。可用的安全防护措施及相关参数在表5.2中给出，一共给出了13 种安全防护措施及其相关属性，与图5.4中的攻击图一一对应。根据第四章节介绍的不确定性赋值方法，各种转移概率的赋值如表5.3所示。



From	To	Protocol/Port
0.0.0.0	Web Server	http/80
	email Server	SMTP/25 POP3/910
	GateWay	SSH/22
GateWay	All Machines in trusted Zone	Basic network protocol
Local user and Admin	GateWay	Basic network protocol
	DB Server	SQL/1433
	File server	NFS/111
	Web Server	http/80
	email Server	SMTP/25 POP3/910
	DNS Server	DNS/53
Web Server	DB Server	SQL/1433

图 5.3: 网络环境拓扑示意图-2

在确定了上述相关的信息后，完整的安全防护策略概率攻击图可被构造出来，如图5.4所示。外部攻击者能通过IIS 的漏洞CVE-2009-1535，攻击Web服务器，如果成功即获得Web 服务器的权限，攻击者可以通过NFS Shell 在Web 数据服务器上写入具有恶意代码的文件，当管理员登录误打开或执行该文件时，即可能被植入木马，攻击者可利用管理员的权限登录进入敏感文件服务器窃取文件。内部攻击者能通过猜密码攻击短暂进入管理员的终端系统，他可以通过种入一个keylogger 程序记录管理员的键盘操作来窃取敏感文件服务器密码，也可能利用管理员的粗心，盗取保存敏感文件服务器密码的私人文档，从而实现对敏感文件服务资料的窃取。

表 5.1: 网络环境系统漏洞和威胁表

Host	Underlying Vulnerabilities or Threats	Comments
Local Desktops 192.168.0.0 – 127	Remote Login Adobe Reader Code Injection Identity Theft	CA 1996-93 CVE-2013-3337 Inside Attack
Admin 192.168.120.200	Identity Theft Privacy Data Theft KeyLogger	Inside Attack Inside Attack Trojan Horse
GateWay 192.168.120.254	Open SSH Heap Overflow	CVE-2003-0693
SQL Server	SQL Injection Attack Anomly of SQL Data Accesses	CVE-2008-0516 Inside Attack
Web Server IIS	Vulnerabilities	CVE-2009-1535
File Server	Identity Theft Large Sensitive Data Downloading Linux nfs-utils xlog() Remote Buffer Overflow	Inside Attack Inside Attack CVE-2003-0252

5.4.2 实验结果分析

据图5.3和表5.2、5.3 确定的概率攻击图$MPAG$，给定一些观测事件和对应的置信概率，并约定代价上限C，利用本章上一节提出的OMSN 算法，可以得到计算得到最优安全防护策略。下面模拟了3组观测事件，展示了如何在概率攻击图上计算最优防护测量的过程，实验结果见表5.4。观测事件序列表示当前观测到的事件及相应的置信度，第2 行是限制代价C，第3 行表示防护前的目标节点攻击概率，第4行表示最优安全策略的选择过程，第5行最优安全策略的实际代价C'，第6行表示防护策略启动之后目标攻击节点的概率取值。测试1 的观测事件序列为空，代表传统安全防护策略的静态计算，即不管当前网络状况如何，根据攻击图上的漏洞依赖关系、专家知识和经验确定转移概率来计算静态的最优安全防护策略。测试2和测试3 都是动态计算最优安全防护策略的例子。测试2 利用OMSN 算法找到了在观测事件序列为$\{o_4 : 0.9, o_{12} : 1\}$ 代价为13 的防护策略集合$\{m_4 = 1, m_5 = 1, m_{12} = 1\}$，使得目标攻击节点的概率从0.843 下降到0.21。图5.5 、图5.6和图5.7 详细地展示测试1、测试2和测试3的计算过程和防护策略的选择顺序，在图中序号1,2,3…表示第1,2,3…次启动安全防护策略后相应路径上每个节点积累概率的变化。图5.6和图5.7中粗黑体字的方块表示测试2、测试3与测试1 不

表 5.2: 安全防护措施及相关参数表

$[m_i, \phi_i, c_i]$	**Explanation of Measures and Related Attributes**
m_1,0.3,9	Password Changing Periodicity and Shortest Password Limitation Effective but High Cost
m_2,0.1,3	KeyLogger Prenvetion Effective and Relatively Low Cost
m_3,0.2,7	Pretection on Privacy Data Effective and Relatively High Cost
m_4,0.5,8	Prevention of Identity Theft Commonly Effective and High Cost.
m_5,0,3,4	Prevention of Sensitive Data Transfering Effective and Relatively Low Cost
m_6,0.05,4	IIS Vulnerability Fix Effective and Relatively Low Cost
m_7,0.8,3	Limitation of NFS Little Effective and Relatively Low Cost
m_8,0.2,8	Protection of File Integrity Effective and Relatively High Cost
m_9,0.3,5	Provention of Trajon Effective and Common Cost
m_{10},0.05,5	Fix OpenSSH Vulnerability Full Effective and Common Cost
m_{11},0.1,6	Fix BufferFlow Vulnerability Full Effective and Relatively High Cost
m_{12},0.1,3	Fix SQL Injection Vulnerabilit. Full Effective and Low Cost
m_{13},0.7,8	Prevention of anomaly on SQL Data Access Little Effective and Relatively High Cost

表 5.3: MPAG的状态转移表

Edge	Value of Probility	Edge	Value of Probility
(s_2, a_1)	0.6	(s_6, a_7)	0.5
(s_0, a_6)	0.5	(a_7, s_7)	1.0
(s_0, a_{10})	0.8	(s_7, a_8)	0.4
(s_2, a_6)	0.9	(a_8, s_8)	1.0
(a_1, s_3)	0.8	(s_8, a_9)	0.6
(s_3, a_2)	0.6	(a_9, s_5)	0.8
(s_3, a_3)	0.8	(a_{10}, s_9)	1.0
(a_2, s_4)	0.9	(s_9, a_{11})	0.6
(a_3, s_4)	0.6	(a_{11}, s_5)	1.0
(s_4, a_4)	1.0	(s_9, a_{12})	0.4
(a_4, s_5)	1.0	(a_{12}, s_{11})	1.0
(s_5, a_5)	0.9	(s_{11}, a_{13})	0.4
(a_5, s_{10})	0.9	(a_{13}, s_{10})	0.8
(a_6, s_6)	1.0	#	#

同的概率计算过程。详细计算过程见图所示。

5.5 本章小结

本章讨论了在概率攻击图上最优安全防护策略的计算方法，分析了最优安全防护策略的多目标优化、静态防护和动态防护等要求，并设计了一套最优安全防护策略算法用于实时计算系统在遭受内部攻击时最好的安全防护策略集合，给出实验和相关分析。

在不考虑防护措施**实施撤销**的代价的情况下，动态安全防护策略的研究可能滑向博弈论的研究。网络管理员根据当前看到的IDS 告警事件，分析当前可能发生着的攻击，并根据攻击发生的可能性和严重程度重新制定安全防护策略，最大化的减轻系统被攻击的危险，保护核心资产。这可能是另一个大的研究课题，即网络信息对抗要研究的一部分问题。实际上，网络攻击具有单调性特征，管理员总是要假定攻击者的学习、探索和攻击过程是一个单调递增的过程，攻击者不会失去已经学到的攻击手段和

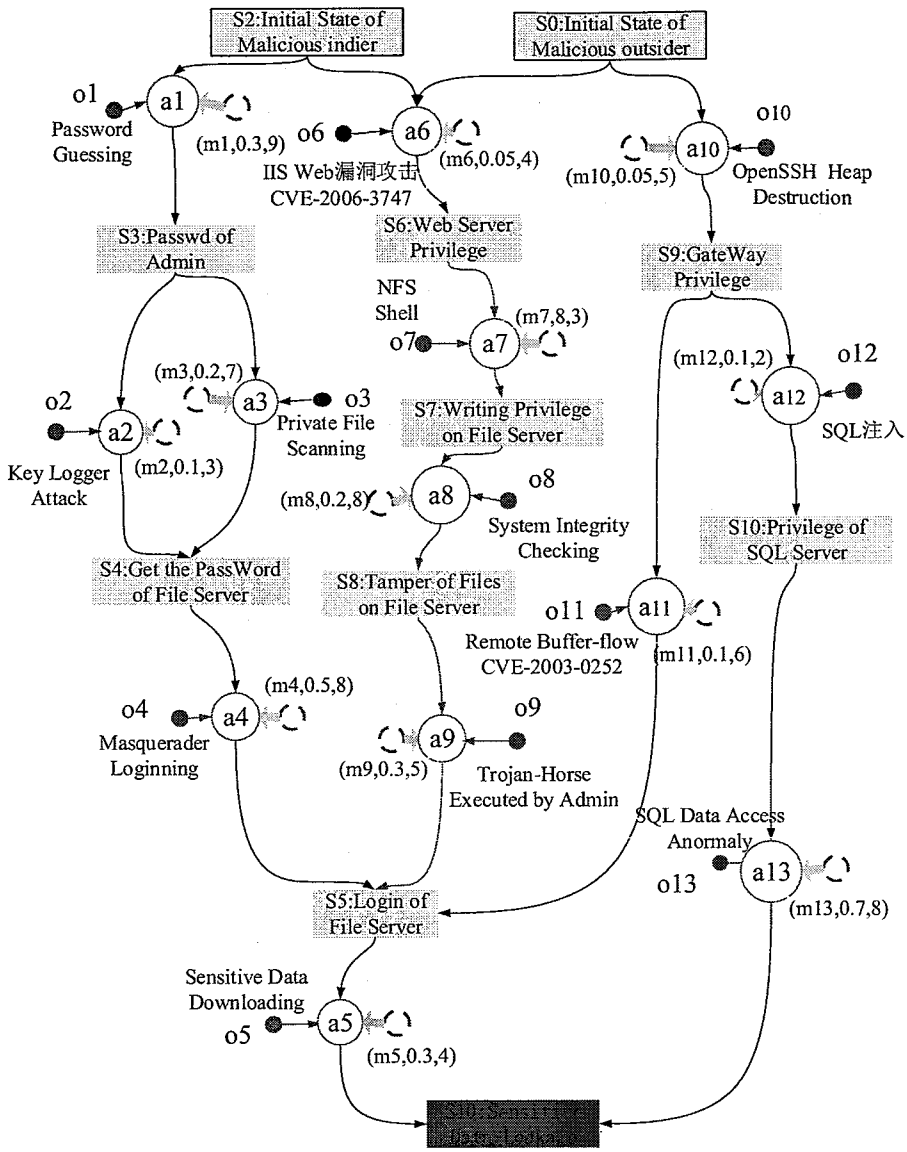


图 5.4: 安全防护概率攻击图实例

已经探测了解到的系统相关知识，也不会无故放弃已经攻破的系统权限。

表 5.4: 最优防护策略计算结果表

Detail of Tests	**Test1**	**Test2**	**Test3**
Event Sequence	null	$\{o_4 : 0.9, o_{12} : 1\}$	$\{o_2 : 1, o_9 : 0.9, o_{10} : 1\}$
Cost Limitation	C=15	C=15	C=18
$P(s_{goal})$ Before	0.457	0.843	0.8328
Selected Hardening Measures	$m_{11} = 1$ $m_5 = 1$ $m_{12} = 1$	$m_4 = 1$ $m_5 = 1$ $m_{12} = 1$	$m_2 = 1$ $m_9 = 1$ $m_5 = 1$ $m_{10} = 1$
Real Cost	C'=13	C'=15	C'=17
$P(s_{goal})$ After	**0.056**	**0.21**	**0.11**

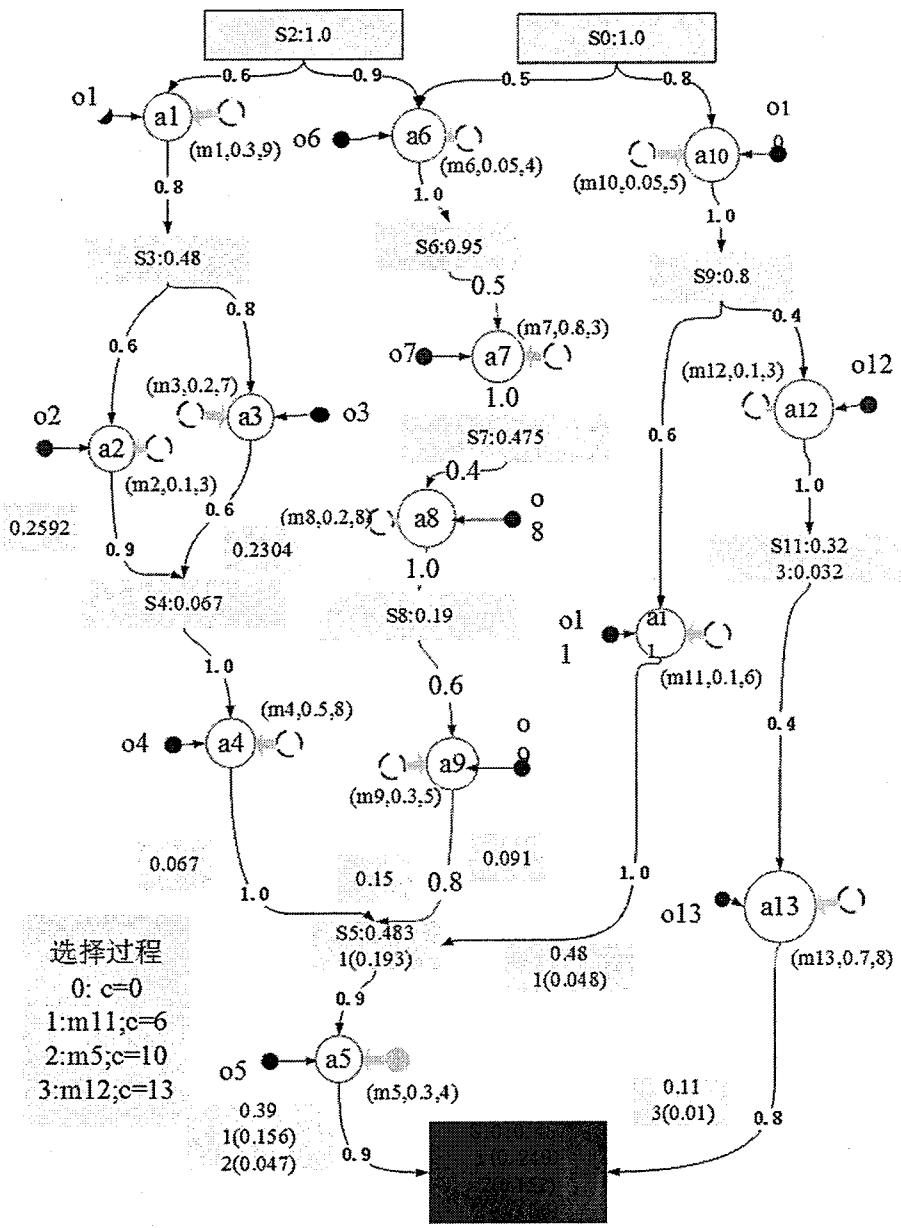


图 5.5: 测试1的最优防护策略计算推导过程

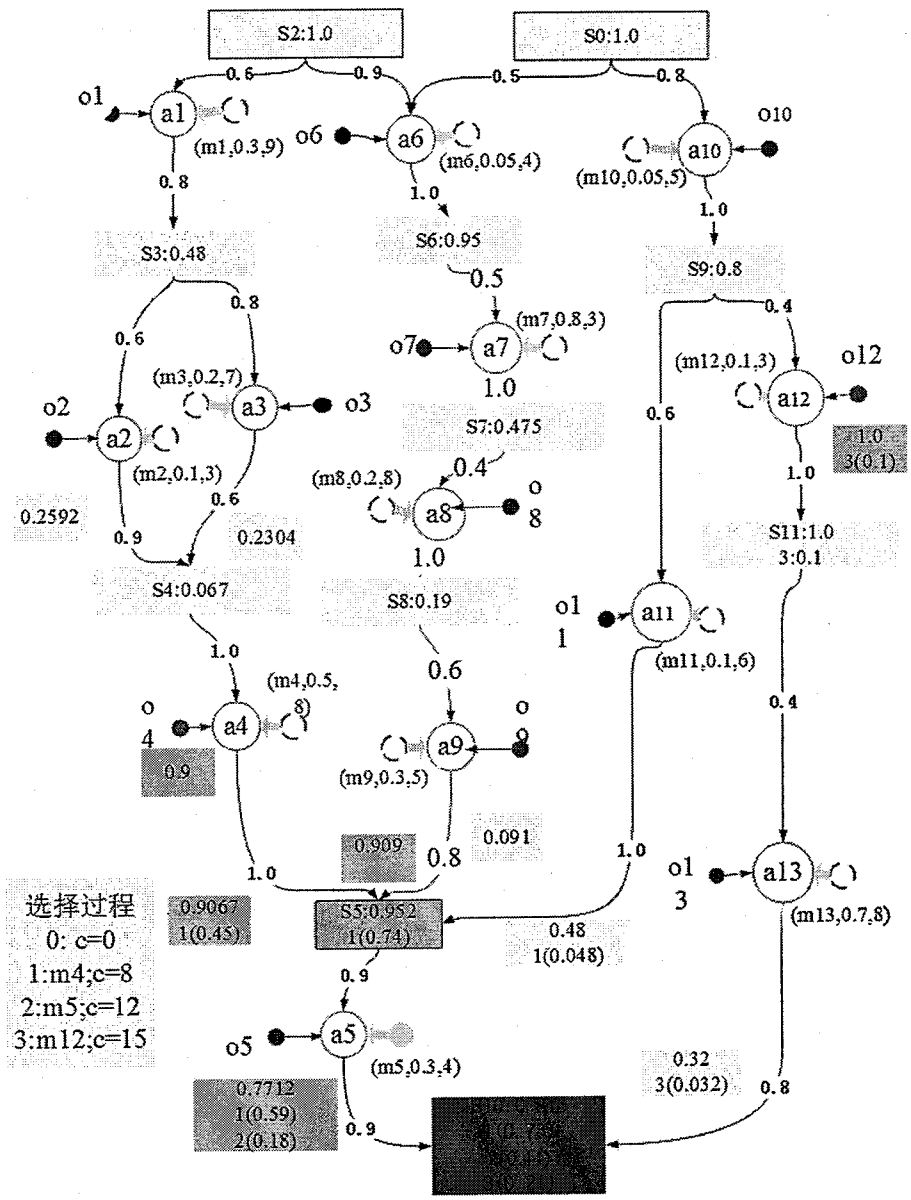


图 5.6: 测试2的最优防护策略计算推导过程

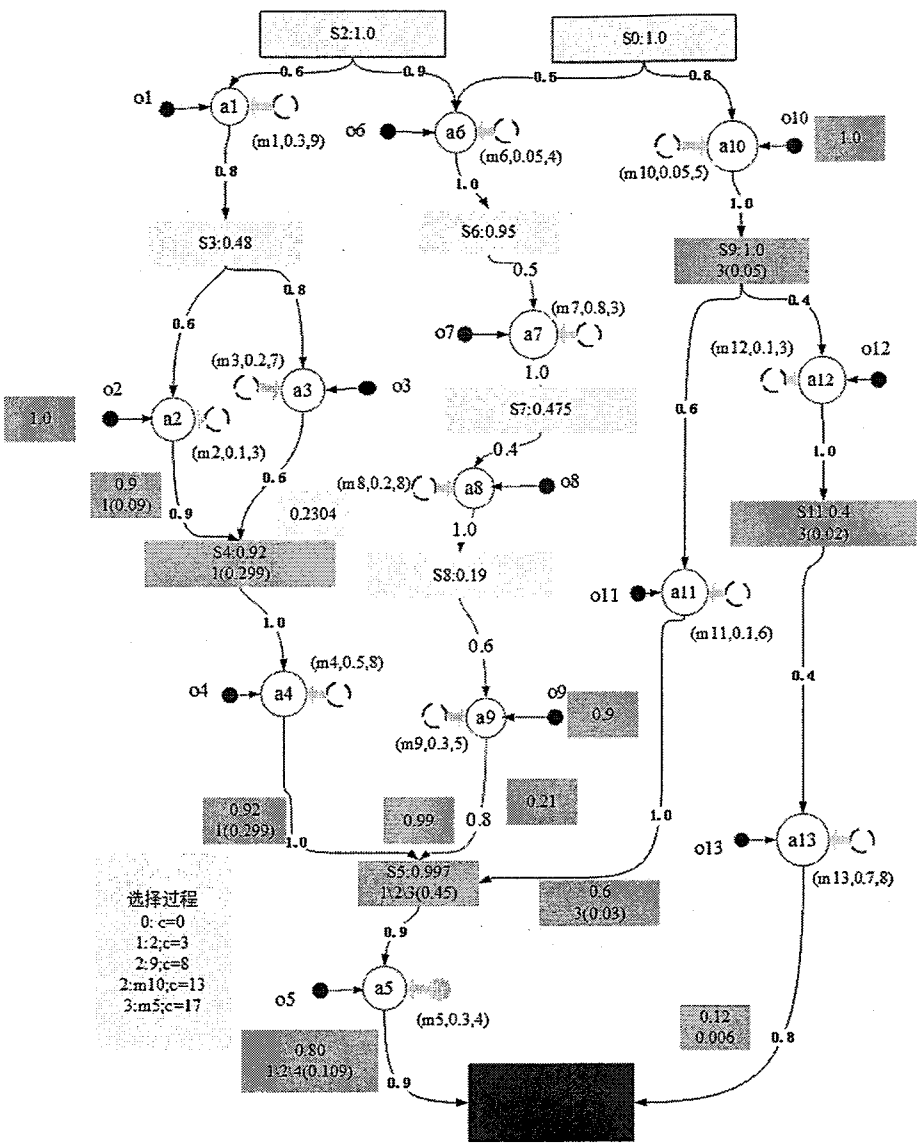


图 5.7: 测试3的最优防护策略计算推导过程

第六章　结束语

6.1　本文工作总结

内部威胁特指来自内部人员的网络攻击，这种攻击行为因为内部人员自然拥有对内部组织和信息系统的相关了解，拥有信息系统的合法访问权限，也拥有接触机密数据的职业便利而变得特别难以检测。更重要的是，内部攻击人员能够合理计划攻击步骤，将攻击动作分解为一步步的小攻击，且每一步小的攻击动作更加隐蔽，更加难以确认，这种情形给传统基于异常检测的方法和基于攻击图的检测方法提出了巨大的挑战。本文在对已有内部威胁检测方法和攻击图技术的研究基础上，对内部威胁的检测框架、异常行为感知，概率攻击图模型及其上面的攻击意图理解，攻击场景重构和最优安全防护策略等问题进行了深入研究。形成的创新性成果和主要贡献如下：

- 深入研究内部威胁检测框架。首先，以人为主线从攻击者的角色、意图和攻击行为产生的观测事件三个大的角度来刻画内部攻击行为的特征，并对内部威胁检测的流程进行梳理，提出了内部威胁异常行为感知、内部威胁攻击意图理解与内部威胁的实时安全防护三个阶段的检测框架，用以指导后面的研究方法和组织。

- 内部威胁异常行为的感知技术主要针对内部攻击过程中可能出现行为异常的检测技术进行研究，包括身份冒用异常、文件访问行为异常和木马心跳通信行为异常检测技术。基于鼠标动力学的身份认证技术提出了一套可实际运行的身份认证系统，在保持认证准确性的同时，将认证时间缩短到秒级，提高了基于鼠标动力学身份认证技术的实用性。文件访问行为检测模型可以检测突发的批量文件访问异常和周期性的文件访问行为，木马心跳通信行为检测能检测连接级和连接内的心跳行为。这些异常行为感知技术比较有针对性解决一些内部异常行为的检测问题，也为后续的意图理解研究提供了数据支撑。

- 内部威胁最大的两个挑战是不确定性和攻击的潜伏性。概率攻击图模型一方面利用攻击图的因果关系描述能力模拟内部攻击的潜伏性，另一方面在攻击图上完备地模拟了内部攻击中可能存在的三种不确定性：攻击者的能力，攻击发生的置信度，攻击成功的概率等。概率攻击图内部攻击具有更强的描述能力。在概率攻击图模型基础上，提出了一种攻击意图推断和场景重构算法，该算法根据观测事件序列和相应的置信度，对目标节点的被攻击概率进行增量计算，给出最大概率攻击路径。实验结果表明，该算法可有效推断攻击意图和重构攻击场景，减少虚警数量。

- 最后，本文提出了一种近似最优的安全防护策略计算算法，该算法通过引入安全防护策略节点及其作用、代价属性，将概率攻击图扩展为安全防护概率攻击图，证明了基于安全防护概率攻击图的最优安全防护策略计算是一个NP难的问题，提出并实现一种贪心算法，可在多项式时间内动态地计算近似最优安全策略。

6.2 下一步研究方向

本文探讨了内部威胁检测技术研究中的一些关键问题，但还远未完整地解决内部威胁，进一步的工作包括：

1. 内部威胁三个影响因素中，攻击者的角色因素没有在后续的研究工作中展开研究，因为攻击者的角色不仅仅与计算机系统相关，也与企业和组织的人事、管理和信息等多个职能部门相关。而本文关注仅从信息系统的角度检测内部威胁，未来的工作可以将信息系统与其他职能部门联合起来解决内部威胁。

2. 在内部威胁异常行为的感知技术中，本文的研究还非常有限。这是一个开放的研究领域，各种可能的异常行为检测算法都应该被深入研究，并加入内部威胁意图理解的数据来源中。

3. 在内部威胁的意图理解方面，本文仅从概率攻击图的角度来研究，获得比较好的理论形式成果。概率攻击图方面存在一个关键的问题是概率攻击图的自动构造。专家知识可以构造已知的攻击模式，而日志关联分析有可能挖掘未知的攻击模式。未来的工作应更多地聚焦在后者上。