**密级： 保密期限：**

xm 拷贝

**硕士学位论文**



**题目：基于安全日志的攻击模式挖掘技术研究**

**学 号： 2014110690**

**姓 名： 李扬**

**专 业： 信息安全**

**导 师： 胡正名**

**学 院： 网络空间安全学院**

**2016年 12月 15日**

独创性（或创新性）声明

本人声明所呈交的论文是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中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。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。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

关于论文使用授权的说明

本人完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。

本学位论文不属于保密范围，适用本授权书。

本人签名： 日期：

导师签名： 日期：

基于安全日志的攻击模式挖掘技术研究

摘 要

如今计算机和互联网技术正在飞速发展，人民在享受着技术进步所带来的各种便利的同时,又不得不面对着越来越严峻的信息安全问题。各种网络安全设备的应用在一定程度上缓解了信息安全问题的威胁。然而,由于攻击事件日渐增多、攻击手段日趋复杂,导致各个设备所产生的安全日志越来越复杂,数量越来越庞大,可读性越来越差。大量的构成威胁的安全日志告警被隐藏在这些海量、零散的安全日志告警之中。通过对安全日志数据进行数据挖掘，挖掘出隐藏其中的攻击模式，还原攻击场景，从而发现攻击的真实意图，能够更加有效的防范攻击行为。

目前安全日志的分析处理面临着许多问题，一方面，算法对专家先验知识有很大的依赖，很难分析出新的攻击行为模式；许多算法只能挖掘出攻击模式的一些片断，在准确率和模式完整性方面有很大的缺陷。另一方面，安全日志数据的数据源逐渐分散，这使得收集原始日志数据变得越发困难。随着网络速度传输加快，网络的流量也越来越大，对安全日志数据的存储查询处理方式却存在瓶颈，导致系统整体性能下降。

针对这些问题，本文进行了深入研究，主要工作如下：

（1）研究了相关的日志收集存储技术和数据挖掘相关理论与算法，为安全日志挖掘积累了理论基础。

（2）提出一种基于改进模糊聚类分析和序列模式的攻击模式挖掘算法。该算法结合了模糊聚类能够描述安全日志间相似性和序列模式挖掘算法能够较好的描述攻击步骤间前后间逻辑关系的优势，可以更加准确的挖掘日志中的隐藏的攻击模式知识。方法首先将安全日志数据根据时间戳转化为全局攻击序列；然后利用改进的模糊聚类分析算法根据安全日志属性之间的相似度，将可能处在同一场景下的序列聚集在一起转化为攻击序列集；最后通过序列模式挖掘算法从攻击序列集中挖掘出攻击者的攻击模式。

（3）设计并搭建了基于ELK的攻击模式挖掘系统。该系统基于ELK技术可以完成从安全日志收集到攻击模式挖掘整个流程。

（4）搭建实验环境对算法进行分析评估，实验结果表明该算法不仅可以有效的挖掘出隐藏在安全日志中的攻击模式，还能在结果中减少无效的攻击模式，生成更有价值的攻击模式知识。

**关键词**：安全日志 序列模式挖掘 模糊聚类分析 攻击模式

ATTACK PATTERN MINING BASED ON

SECURITY LOG

ABSTRACT

Nowadays, computer and Internet technology are developing rapidly. People enjoy the convenience brought by technological progress and have to face more and more serious information security problems at the same time. The application of network security devices have used to defense network attacks. However, due to the increasing number of attacks and the increasing complexity of attack methods, the security logs generated by various devices are becoming more and more complicated. The number of them is getting larger and larger, and the readability is getting worse and worse. A large number of threats is hidden in these massive security logs. Through the data mining of the security logs, the hidden attack mode is unearthed and the attack scene is refactored. So that the true intention of the attack can be found, and the attacking behavior can be more effectively prevented.

At present, the analysis and processing of security log is facing many problems. On one hand, the algorithm has a great dependence on experts' prior knowledge, and it is difficult to analyze new attack patterns. Many algorithms can only dig out some fragments of attack patterns. The accuracy of mining results is very low. On the other hand, data source of the security log gradually dispersed, which makes the collection of raw log data becomes more difficult. With the speed of network transmission speed up, the network traffic is also growing, the security log data storage query processing methods are bottlenecks, resulting in overall system performance.

In view of these problems, this paper has carried on the thorough research, the main work as follows:

(1) The related log collection and storage technology and data mining related theories and algorithms are studied, which accumulates the theoretical foundation for security log mining.

(2) This paper proposes an attack pattern mining algorithm based on improved fuzzy clustering analysis and sequence pattern. This algorithm can describe the similarity between the security log and the sequence pattern mining algorithm, and also can describe the logic relation between the attacking steps well, which make it could excavate the hidden attack pattern knowledge in the log more accurately. Firstly, the security log data is transformed into the global attack sequence according to the timestamp. Then, the improved fuzzy clustering algorithm is used to aggregate the sequences that may be in the same scene according to the similarity between the security log attributes. Finally, the attack pattern of the attacker is extracted from the attack sequence by the sequential pattern mining algorithm.

(3) Design and build an attack pattern mining system based on ELK. The system is based on ELK technology can be completed from the security log collection to the attack mode mining.

(4) Set up the experimental environment to analyze and evaluate the algorithm. Experimental results show that the algorithm can not only effectively exploit the attack patterns hidden in the security log, but also reduce invalid attack patterns in the results and generate more valuable attack pattern knowledge.

KEY WORDS：security log, sequence pattern mining, fuzzy clustering analysis, attack pattern

目录

[第一章 绪论 1](#_Toc470860544)

[1.1课题背景和意义 1](#_Toc470860545)

[1.2国内外研究现状 2](#_Toc470860546)

[1.3研究目的和内容 4](#_Toc470860547)

[1.4论文结构 4](#_Toc470860548)

[第二章 相关理论与关键技术 6](#_Toc470860549)

[2.1 安全日志概述 6](#_Toc470860550)

[2.2 黑客攻击步骤 7](#_Toc470860551)

[2.3 数据挖掘概述 8](#_Toc470860552)

[2.3.1 数据挖掘定义 8](#_Toc470860553)

[2.3.2 数据挖掘过程 8](#_Toc470860554)

[2.4 常见数据挖掘分析方法 9](#_Toc470860555)

[2.4.1 关联分析 9](#_Toc470860556)

[2.4.2 序列模式分析 11](#_Toc470860557)

[2.4.3 分类分析 13](#_Toc470860558)

[2.4.4 聚类分析 14](#_Toc470860559)

[2.5安全日志收集与存储技术 15](#_Toc470860560)

[2.5.1 基于HDFS的文件存储技术 15](#_Toc470860561)

[2.5.2 Logstash日志采集技术 17](#_Toc470860562)

[2.5.3 ElasticSearch日志检索技术 17](#_Toc470860563)

[2.5.4 Kibana日志可视化分析技术 19](#_Toc470860564)

[2.5.5 Redis数据库技术 20](#_Toc470860565)

[2.6 本章小结 21](#_Toc470860566)

[第三章 基于模糊聚类分析和序列模式的攻击模式挖掘算法 22](#_Toc470860567)

[3.1 安全日志特点分析 22](#_Toc470860568)

[3.2基于模糊聚类分析和序列模式的攻击模式挖掘算法流程 24](#_Toc470860569)

[3.3 安全日志序列集生成 25](#_Toc470860570)

[3.3.1 模糊聚类分析概述 25](#_Toc470860571)

[3.3.2 安全日志属性相似度函数 27](#_Toc470860572)

[3.3.3 模糊聚类分析算法改进 30](#_Toc470860573)

[3.4 基于序列模式的频繁集挖掘 31](#_Toc470860574)

[3.5实验分析 33](#_Toc470860575)

[3.5.1 实验数据集 33](#_Toc470860576)

[3.5.2 实验环境 34](#_Toc470860577)

[3.5.3 算法执行时间对比 34](#_Toc470860578)

[3.5.4 算法结果对比 35](#_Toc470860579)

[3.6 本章小结 37](#_Toc470860580)

[第四章 基于ELK技术的攻击模式挖掘系统 38](#_Toc470860581)

[4.1 系统概要与整体架构 38](#_Toc470860582)

[4.2 主要功能模块 39](#_Toc470860583)

[4.2.1 安全日志采集 39](#_Toc470860584)

[4.2.2 安全日志处理 41](#_Toc470860585)

[4.2.3 安全日志存储 44](#_Toc470860586)

[4.2.4 攻击模式挖掘 45](#_Toc470860587)

[4.2.5 系统监控 46](#_Toc470860588)

[4.3 本章小结 47](#_Toc470860589)

[第五章 实验与分析 48](#_Toc470860590)

[5.1 实验环境搭建 48](#_Toc470860591)

[5.2 实验过程 49](#_Toc470860592)

[5.2.1 安全日志收集测试 49](#_Toc470860593)

[5.2.2 挖掘结果对比 52](#_Toc470860594)

[5.3 本章小结 54](#_Toc470860595)

[第六章 总结与展望 55](#_Toc470860596)

[6.1 本文工作总结 55](#_Toc470860597)

[6.2 未来展望 55](#_Toc470860598)

[参考文献 57](#_Toc470860599)

[致 谢 60](#_Toc470860600)

[攻读学位期间发表的学术论文 61](#_Toc470860601)

第一章 绪论

1.1课题背景和意义

随着计算机网络和通讯技术的飞速发展，互联网变得越来越重要。2015年国家互联网+战略开始实施，互联网的建设也已经成为了我国基础设施建设中不可或缺的一部分。人类的生活与互联网变得更加紧密不可分割，随之而来的计算机网络安全问题也更加深刻的影响着整个国家和社会的稳定。根据CNCERT《2015年我国互联网网络安全报告》[1]2015年仅CNCERT/CC就接收到国内外相关的网络安全攻击事件126916次，比2014年增长了125.9%。国内的攻击事件达到126424起，比2014年增长128.6%。2015年捕获恶意程序总量为112589个，比2014年增长6.5%。日均DDos攻击事件1558起，比2014年增长21.4%。可见，由于网络运行的规模逐渐扩大，网络攻击的攻击范围和攻击频率也显著增加。现代社会对计算机网络技术的依赖不断加深，网络安全相关问题更加值得关注。不论是微观的个人电脑用户还是宏观上的大规模网络设施中，信息安全问题已是计算机领域无法逃避的问题。

为了防范网络攻击，许多网络安全设备得到了非常广泛的应用，例如入侵检测系统（Intrusion Detection System，IDS）和网络安全防火墙（Firewall）等。这些系统或软件可以对大部分的网络单步攻击行为进行告警处理，能够一定程度上遏制攻击者的攻击行为，监控网络的运行状况。但是随着互联网的发展，入侵行为的数量急速增加，这些安全设备产生的日志数量也急剧扩大，最后呈现给安全管理员的经常是数量庞大的安全日志告警信息。同时，日志中的告警信息也往往是孤立的，这些安全警报日志是由某一种网络行为所引起的，用于反映某一种网络中的异常行为（例如一次登录失败，端口扫描，获取root权限等）。如果一次攻击行为仅为一次攻击企图，其目的只是为了后续的其他攻击步骤做准备，并不代表着真正的攻击，那么仅仅从这条安全日志中就很难得到有价值的信息。只依靠安全管理员通过人工的方式从这些低级别孤立的安全告警日志中，发现攻击者深层次的攻击行为和攻击意图变得十分困难。另外，随着大数据时代的到来，安全设备也逐渐应用在分布式环境中，对这种大规模格式不统一的安全日志进行统一的收集管理也变得越发困难。这些问题的存在严重地制约了网络安全管理人员对攻击的响应处理效率。因此，如何对这种大规模的安全日志告警信息进行收集，并分析隐含于其中的攻击模式成为了安全分析领域的研究热点之一。

网络中的多步攻击往往会使用各种攻击手段来达到攻击目的，具有复杂多变的行为特征。例如DDos攻击事件，为了达到攻击目的，攻击者会采用不同的方式控制大量的含有不同漏洞的主机，攻击手段和模式千变万化[2]。安全日志中记录的安全告警事件正是对这种网络攻击行为模式的反映。一个攻击者的攻击行为或手段在一定程度上必然会反映在网络设备的安全日志中。面对海量的安全告警日志，使用数据挖掘技术对其中隐藏的黑客的攻击模式进行提取分析，发现安全日志间告警的关联信息，不但可以在收到网络攻击后对攻击的行为场景进行重现、追踪溯源，也能为防范此类模式的攻击行为提供强有力的支持。

本文将数据挖掘技术与安全日志关联分析相结合，提出了对网络安全日志中隐含的攻击行为序列模式进行挖掘。挖掘出的攻击序列模式揭示了大多数攻击者对网站进行攻击所对应的攻击步骤。这些信息一方面为安全管理员重建攻击场景，了解黑客的攻击行为模式提供参考；另一方面为入侵检测系统提供新的关联模板或关联规则，从而为网站管理员提供更高级别的安全告警信息。

1.2国内外研究现状

在此之前，很多研究人员已经提出了许多通过对安全告警日志关联分析，挖掘攻击者攻击模式的方法。本文总结为以下三类：基于因果关系的安全日志告警关联分析方法，基于机器学习的攻击模式发现方法以及基于数据挖掘的攻击模式挖掘方法。

(1) 基于因果关系的告警关联

Templeton[3]等在安全日志分析中首先提出了基于因果关联对安全日志进行分析的方法。该方法首先对安全日志告警之间的前因后果关系进行分析，定义每一类攻击的前提和结果，然后利用分析出的先决条件对两个安全日志告警进行关联。在此过程中作者提出了JIGSAW的结构语言来对攻击者的攻击行为模式进行描述。北卡罗莱纳州大学计算机防御实验室的Peng Ning等[4]在对这种方法深入的研究基础上提出了TIAA安全日志分析方法。该方法也是根据人工定义的两个攻击步骤之间的前因后果对安全日志中隐含的攻击场景进行构建。在此之后，Cuppens F[5,6,7]等 也使用了相似的关联方法。在MIRADOR项目中对于IDMEF格式的告警信息，使用Prolog谓词逻辑对攻击行为的关联逻辑进行描述。他们认为多步攻击的行为模式之中，一个攻击步骤不是独立的，前期的行为是为后期做准备。他们利用这种攻击步骤间的因果关系定义出自动关联规则，然后通过制定的规则将安全日志的告警信息关联起来，形成攻击模式信息。

以上这些算法都是属于根据人工事先定义的因果关系规则对安全告警日志进行分析的算法。这类算法的关键在于如何对关联规则进行定义。而对因果关联规则的定义过程中，需要大量的专家知识，因此设计难度大，因果关系复杂，在实际应用中很难实现。当两个攻击步骤之间不存在明确的因果关系时，很难对这两个告警日志定义关联规则。由于这些大量的关联规则完全是以专家的先验知识为基础的，因此只能对已经存在的攻击模式进行发掘与重现，难以发现新的攻击场景和攻击模式。

针对这种因果关联的方法难以发现新的攻击模式的问题，在前人基础上Lee Wenke [8,9,10] 等提出了基于统计时序来进行攻击模式片段挖掘的方法。对关联出的攻击场景片段进行连接，形成完整的攻击行为模式。该方法提出了Granger 因果指数（GCI，granger causality index）的概念。通过时间序列模型来计算GCI指数。根据GCI指数的大小来判断两个安全日志之间是否含有Granger 因果关系。虽然该方法在定义因果关联知识的过程中不需要大量的专家先验知识，并且可以发现新的多步攻击模式。但是在攻击模式片段连接过程中则需要很多专业领域的知识，同时由于发现的是攻击模式片段，这极易造成攻击模式的割裂，很难发现完整的攻击模式知识。

（2）基于机器学习的攻击模式发现方法

GHORBANI 等[11]在安全日志分析过程中使用了机器学习的方法，通过多层感知器和支持向量机（Support Vector Machine，SVM）来进行安全日志攻击模式的发现。Kavousi F等[12]提出使用改进的贝叶斯分类器对安全攻击日志进行关联，生成攻击模式。机器学习的方法首先利用安全日志训练集对模型进行训练，然后利用训练出的模型进行攻击模式的发现。根据他们的算法过程可以看出，类似的机器学习方法需要使用训练数据，训练数据的质量某种程度上决定了训练模型的质量，进而影响到了攻击模式发现的效果。但是在如今的安全领域中，寻找到符合要求并且数量足够的训练数据集是很困难的。因此将机器学习算法应用在安全日志攻击模式的发现过程中有一定的局限性，目前这方面的研究也较少。

（3）基于数据挖掘的攻击模式挖掘算法

随着数据挖掘技术的快速发展，越来越多的研究者逐渐意识到了其在安全日志分析中的重要作用。Ai-fang Zhang等[13]提出使用数据挖掘中关联分析算法来进行攻击模式挖掘的方法，该算法首先使用滑动时间窗口划分警报子序列，然后对安全日志警报子序列进行关联分析发现安全日志中隐含的攻击模式。Hellerstein等[14]提出了从历史安全日志中挖掘突发事件，周期事件和相互依存事件的算法。James J等[15] 提出了使用Apriori 算法发现IDS安全告警日志间关联关系的算法。现有的方法在挖掘结果的准确性方面还有欠缺。由于准确率的不足，在对结果的分析时存在一定的困难。同时，有些方法[13]在挖掘时仅仅考虑了时间因素，这极易造成攻击模式挖掘结果的场景分裂，影响了方法的实用性。

1.3研究目的和内容

本论文的目的是，利用数据挖掘技术从海量的安全日志数据中挖掘出黑客的攻击模式，深入解析安全日志告警类型间存在的关联关系和黑客的攻击步骤间的发生模式，从而对攻击者的攻击模式进行更加完整的重现，为安全管理员防范网络攻击和入侵检测系统的关联规则制定提供更加丰富可靠的攻击模式知识。为此本文的具体研究内容和成果如下：

（1）研究了相关的日志收集存储技术和数据挖掘相关理论与算法，为安全日志挖掘积累了理论基础。

（2）提出一种基于改进模糊聚类分析和序列模式的攻击模式挖掘算法。该算法结合了模糊聚类能够很好的描述安全日志间相似性和序列模式挖掘算法能够较好的挖掘攻击步骤间前后间逻辑关系的优势，可以更加准确、完整的挖掘日志中隐藏的攻击模式知识。方法首先将安全日志数据根据时间戳转化为全局攻击序列；然后利用改进的模糊聚类分析算法根据安全日志属性之间的相似度，将可能处在同一场景下的序列聚集在一起转化为攻击序列集；最后通过序列模式挖掘算法从攻击序列集中挖掘出攻击者的攻击模式。

（3）设计并搭建了基于ELK（ElasticSearch Logstash Kibana, ELK）的攻击模式挖掘系统。该系统基于ELK技术可以完成从安全日志收集到攻击模式挖掘整个流程，同时详细分析了各个功能模块的实现方案。

（4）搭建实验环境对算法进行分析评估，实验结果表明该算法不仅可以有效的挖掘出隐藏在安全日志中的攻击模式，还能在结果中减少无效的攻击模式，生成更有价值的攻击模式知识。

1.4论文结构

第一章为绪论。介绍了国内外严峻的信息安全形势，分析了安全日志分析领域的研究现状和所面临的问题，指出研究基于安全日志的攻击模式挖掘技术的必要性，并简要说明了论文结构。

第二章为相关理论与关键技术。主要研究了日志收集存储相关技术和数据挖掘相关理论及其相关挖掘算法，以及对安全日志和攻击模式进行了简单分析，为下面的算法提供了理论基础。

第三章为基于改进模糊聚类分析和序列模式的攻击模式挖掘算法。算法首先对安全日志的特点进行分析，在此基础上提出使用改进的模糊聚类分析生成序列集之后，再利用序列模式挖掘算法挖掘频繁集的思想来进行攻击模式的挖掘。本章主要阐述了算法的计算过程，重点对安全日志属性间相似度函数的计算方法进行分析阐述。最后利用仿真数据集进行实验来验证本算法的有效性。

第四章为基于ELK技术的攻击模式挖掘系统。本章首先介绍了系统的整体架构，然后分别对各个功能模块的设计实现进行介绍。

第五章为实验与分析。本章搭建系统实验环境，对基于ELK技术的攻击模式挖掘系统进行测试，并对系统收集的真实安全日志数据进行攻击模式挖掘，验证系统的稳定性和算法的准确性。

第六章为总结与展望。概括总结了本文的主要研究成果和不足，对未来的可研究方向进行了展望。

第二章 相关理论与关键技术

本章首先介绍了安全日志相关概念，并对黑客的攻击模式进行了分析总结；然后重点研究了数据挖掘领域的相关理论与算法；最后介绍了从海量的安全日志中挖掘攻击者攻击模式过程中涉及到的相关技术，主要从安全日志的收集检索和数据挖掘技术两个方面进行介绍。主要介绍了HDFS存储技术、基于ELK、Redis的日志收集检索技术。

2.1 安全日志概述

日志，指的是对设备运行行为的记录，是设备对某对象处理过程和处理结果的记录按照时间排序后的集合[16]。在计算机安全领域中，安全日志指的是对网络和计算机等进行安全监控的相关安全设备对敏感行为进行的记录。产生安全日志的设备有很多，主要包括防火墙、杀毒软件、IDS等。这些安全设备在监控运行过程中，会不断的产生反映网络安全行为的各种日志信息。安全管理人员可以利用这些安全日志对所管理的网络或计算机系统进行有效的监控，同时还可利用历史的安全日志来分析更深层次的安全态势。

在现实应用情境中有存在多种网络安全日志类型，其中最主要的是IDS日志，通过在IDS系统中制定相应的匹配规则或者策略，可以监控网络或计算机系统中流量的异常信息，进而发现可能的攻击行为[17]。IDS从行为模式上来说是一种相对主动的网络安全防护技术。IDS在对网络的攻击行为进行有效监控时对网络性能的影响很小。分析IDS告警日志也是安全管理员对网络安全管理的一种非常有效的手段。和其他的安全保护设备相比，IDS的检测相对比较智能化。合理的制定规则和防御识别策略可以很大程度的减少安全管理员的工作强度。当然，在一般情况下为了保证整个网络的安全，入侵检测系统的策略制定都相对严格。因此，对于入侵检测系统来说，其日志的产出量是十分巨大的[18]。在大型的网路系统中每天会产生数十GB的IDS日志，这给网络的安全管理效率造成了很大的影响。本文所研究的从安全日志中挖掘攻击模式的方法正是对这种海量安全日志进行高效利用的众多手段之一。

2.2 黑客攻击步骤

由国家互联网应急中心（CNCERT/CC）《2015年我国互联网网络安全报告》[1]中对网络安全攻击的攻击类型统计中能够发现，80%的攻击事件都是包含许多个攻击步骤的。黑客在对目标网络进行攻击时，一般是通过多种威胁性较低的攻击步骤逐渐深入攻击，最终达到黑客的攻击目的。一次完整的攻击会经过一系列的攻击步骤，之前的攻击往往是对后面更加深入的攻击所做准备。比如，如果黑客要入侵某网络，那么他就需要先通过端口扫描的步骤来发现目标网络中存活的主机，然后通过主机的开放端口来对目标主机进行下一步骤的攻击行为。这种黑客为了达到攻击的最终目的，而进行的一系列攻击步骤所组成的攻击序列就是黑客的攻击模式。一般来说黑客的攻击有以下几个步骤：

(1)目标信息收集

进行攻击前首先要了解攻击目标的各种信息。知道IP地址只是信息收集过程中的第一步。一个成熟的黑客对一个大型网络进行攻击时，应该能够将该网络的大体框架进行了解，分析他们所要攻击目标所处的网络环境。根据这些网络信息资料对目标网络进行扫描，比如扫描端口，操作系统版本类型检测等。通过扫描不仅可以发现网络内存活的正在通信的主机，还可以找到可以利用的软件漏洞。信息收集常用的协议或工具有SNMP协议、TraceRoute程序、Whois协议、Ping命令等。这个过程的危害性比较小，目标网络虽然会对这些操作进行记录，但是这些记录会被其他海量的正常信息所掩盖。

（2）获取权限

当对目标网络收集完信息后，攻击者就会利用这些信息来获取目标网络中的某些设备的登录权限。针对不同的系统攻击者会采用不同的方式，比如尝试弱口令登录，盗取共享的敏感文件，协议欺骗，网络监听等等。攻击者如果要对主机进行入侵，仅仅获取普通的用户权限还不算成功。由于对普通用户的权限限制比较严格，黑客很难实施后续的有效攻击手段。这就需要对使用权限进行提升，进而获得超级用户权限。获得超级权限的方式也有很多种，比如通过密码破解，网络监听，还有通过缓冲区溢出漏洞等方式。当攻击者获得超级管理员权限后，便可对所入侵主机进行全面掌控。

（3）实施攻击

攻击者控制目标主机后，就可以利用所掌握的权限对目标主机进行控制。攻击者可以使用多种攻击方式：安装特洛伊木马，为以后长时间的监控主机行为打下基础；篡改盗取敏感的数据信息，获取有价值的数据如银行卡密码等；将控制的主机作为跳板主机对网络内的更多主机进行更进一步的渗透攻击，最终完成攻击目的。

（4）消除痕迹或安装后门

攻击成功后,攻击者可能会对攻击过程中产生的攻击痕迹进行篡改或消除达到隐匿攻击者身份的目的，比如删除系统中事件日志记录、隐藏文件等。有时攻击者为了方便下次攻击，也可能在系统中留下后门。

黑客的攻击总体上可以分为上述四个方面，有时攻击者可能不会采取所有的攻击步骤，同时为了更好的入侵，也有能会使用更多更隐蔽的攻击手段。但是可以看出，攻击者的攻击步骤之间在时间上是有一定的前后因果关系，有些攻击步骤如果单独分析是很难发现有价值的信息。因此挖掘黑客的攻击模式能够很好的帮助安全管理员了解黑客的攻击意图，提高对多步攻击的防范能力。

2.3 数据挖掘概述

从1946年到现在，计算机技术已经经历了60年的发展，在这过程中存储在数据库中的各种信息不断的积累增加。与此同时，伴随着信息时代的到来，人们获取各种信息的手段不断的进步，整个人类社会已经进入了“大数据”时代。数据库中存储了海量的数据，如何利用这些数据进行分析挖掘，成为计算机界日益关注的问题。为了实现这种目的，数据挖掘（Data Mining）技术应运而生。

2.3.1 数据挖掘定义

数据挖掘也被人们称为发现数据库中的知识，就是根据事先设定的目标，从海量的、不完整的、有噪声的、随机的数据集中，提取那些被隐含在数据集中的、人们之前并没有意识到的、将来可能在某些情况下有用的相关知识的过程[19]。数据挖掘不仅限于计算机领域，它是一个新兴的交叉学科，包含了数学、专家系统等多种相关知识。它的应用范围也广泛，只要该领域能够产生并收集到足够的数据，那么数据挖掘就可对该领域的发展发挥出极大的推动作用。数据挖掘发现的知识分为广义知识、关联知识、分类知识、预测型知识、偏差型知识等。

2.3.2 数据挖掘过程

数据挖掘可以处理的数据对象的范围很广，不论是数据库、文件仓库还是其他信息资源库，它都能从中挖掘出很多有意义的隐藏知识。狭义上来说，数据挖掘仅仅是利用算法从数量庞大的数据中挖掘出隐藏其中的相关知识信息，但是在实际使用中数据挖掘包含的范围非常广泛，它是一整套数据处理流程，包括数据准备、挖掘数据和挖掘结果的表达和评价等。数据挖掘整个过程可以归纳总结为下列步骤[20]：

(1)数据的预处理：对数据进行预先的处理，消除数据中的噪声和不完整的数据，同时对数据格式或类型进行变换形成统一的规范化数据，提高数据质量。

(2)数据收集：在此步骤中，通过相应的手段将各种预处理后的相关需求数据进行统一的集成处理，存储到统一规划的数据库中。

(3)数据挖掘：这是数据挖掘过程中的核心步骤。在此步骤中，需要通过数据挖掘相关的算法，对预处理后存储在持久化服务器中的数据进行计算挖掘，从数据中分析其中隐含的数据模式特征或相关知识，并将挖掘结果存入数据库。

(4)结果评价：对挖掘算法产生的数据进行分析评价。分析出挖掘结果所展示的具体知识信息和其中有价值的信息。

根据数据挖掘过程的相关步骤，一个数据挖掘系统的主要组成如图2-1所示：

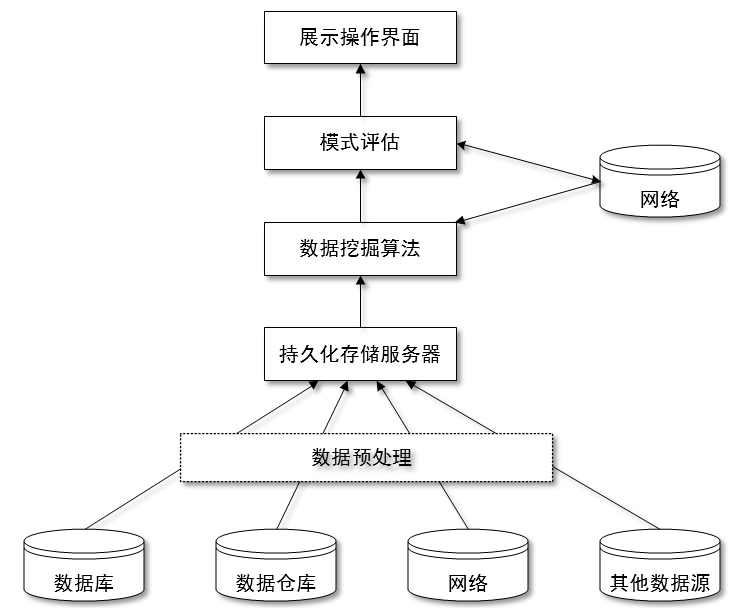


图 2-1 典型数据挖掘系统模型

2.4 常见数据挖掘分析方法

2.4.1 关联分析

在1993 Agrawal等人首先提出了关联分析（Association Analysis）算法。关联分析算法就是通过相关算法挖掘出项目之间潜藏在数据里的相互关联关系[21]。如果给定一组对象和这种相关对象的集合记录，那么就可以通过记录集合进行关联分析总结出在此集合中对象之间的关联关系。

关联规则的发现主要步骤包含三个：数据连接，设定最小支持度（Mini Support）和最小置信度（Mini Confidence），挖掘结果的评估。事务数据库是关联分析的主要对象。关联分析的最终目的是寻找出海量数据集中各种属性间的潜在联系。关联分析最后的结果评价一般使用以下四个指标（假设 是一个关联规则，并且满足）：

（1）置信度(Confidence)：

 （2-1）

由公式（2-1）可知，置信度实际上是在X发生时Y发生的条件概率。置信度是对生成关联规则准确度的描述。当置信度越高时，那么在一条事务中X存在时，Y存在的可能性也就越大。

（2）支持度(Support)

 （2-2）

支持度表示了X和Y同时发生的概率，如果支持度越高那么X和Y同时在同一交易中发生的概率就越高。支持度表示了关联规则在数据集中出现的频繁程度，是对关联规则评价的一项十分重要的指标。

（3）期望可信度(Expected confidence)，期望置信度表示在没有X出现的情况下那么Y出现的情况统计。

（4）作用度(Lift),作用度表示X对Y的影响力大小。只有作用度大于1时才能说明关联规则是有意义的。

在以上四个指标中，支持度以及置信度可以相对直接的对关联规则的具体性质进行描述。在实际使用中，通常只有超过相应数值的支持度和置信度的关联规则才有意义。这种满足条件的关联规则被称为“强规则”，忽略这两个属性会在关联分析过程中产生无数的关联知识。因此，在关联分析的过程中需要制定两个参数：最小支持度（Mini Support）和最小置信度(Mini Confidence)。满足最小支持度的关联规则代表了一组物品集在统计学概率上发生的最小概率；满足最小置信度的关联规则表示其关联规则满足了最小可靠度。因此，关联分析的基本任务就是:从给定的某个事务数据库中，通过关联算法找出满足Mini support和Mini Confidence的关联规则。为了实现任务，关联分析的过程主要有以下两个步骤：

（1）遍历出所有的能够满足最小支持度的所有项集

（2）从第一步获取的所有项集中寻找满足最小置信度和支持度的关联规则。

常见的关联规则算法有Apriori, AIS, DHP等。关联分析能够帮助研究者寻找出潜藏在数据中的项集关系。虽然将关联分析挖掘算法应用于攻击模式挖掘可以发现一定的攻击模式知识，同时也能发现新的攻击知识。但是，由于网络的动态性，数据集中的数据数量随着时间不断增长，不同时间范围内的项目之间的关联强度是不同的。两个项目之间的关联关系不仅仅决定于支持度和置信度，同样还由项目之间的时间所决定。由于在关联分析中每一项元素的权重都是一样的，时间并没有被关联分析考虑在内，因此当前的关联规则并不能解决安全日志模式挖掘过程中的这种问题。另外，当挖掘的数据集较大时，即使采用了优化策略，关联分析算法在效率上仍然存在瓶颈。

2.4.2 序列模式分析

序列模式分析（Frequent Episode Analysis）和关联分析方法比较类似，同样是从交易的事务数据库中寻找频繁发生的数据项之间的关系。两者之间的区别是：序列模式分析的每一条交易是按照时间先后顺序排列成交易序列的。分析过程中强调的是分析的事数据项之间的先后顺序关系。序列模式的应用场景是：将给定的事务数据库中的每一个序列按照时间顺序进行排列，多条序列共同构成一组序列集；然后，通过相应的序列模式挖掘算法，对该序列集进行频繁序列的挖掘；最后，返回大于最小支持度（Mini Support）的子序列[22]。

为了挖掘出频繁序列，序列模式分析的挖掘过程主要包含五个具体步骤[23]：

(1)数据集排序 (Data Sort)

对给定的数据库，将其中的元素按照某种属性进行排序，形成序列数据库。比如将安全日志按照产生的时间排序，这样最后分析的序列结果是与安全日志的产生时间有关。

(2)搜索大项集 (Litemset)

在时间序列数据库中寻找1-序列（长度为n的序列记为n-序列）中满足最小支持度的频繁1-序列，生成频繁1-序列集（也叫1-序列模式）。

(3)转换(Transformation)

为了快速判断序列中是否包含大序列集合，需要将原始交易数据库的属性使用另一种形式替代。例如将字符串形式转换为数值型。同时转换阶段完成之后，交易序列数据库中的每条交易序列都包含了大项集，如果一条交易序列没有包括任何一个大项集，那么这条交易将被舍弃，在后续的迭代寻找中将不会考虑。但是在计算交易数据库中的交易序列的总数时，它将被包含在内。

(4)序列模式(Sequence Pattern)

根据第二步的1-序列模式，使用相关的算法对所有的频繁子序列进行迭代查找。

(5)最大序列(Maximal) （此过程可以省略）

从频繁序列的集合中找出长度最大的子序列(Maximal Sequences)。

根据（4）（5）阶段的不同，序列模式分析的方法有很多，常见算法如下：

(1)Apriori All算法

Apriori All 算法和Apriori算法类似，将Apriori的基本挖掘思想应用到了序列数据库中。首先他先利用大项集阶段产生的1-序列作为种子集，利用这个种子集，穷举产生可能的序列。然后遍历数据库，确定满足最小支持的频繁序列，以此序列为种子集不断的迭代。Apriori All 虽然算法简单，但是缺点很多。算法的每一次迭代都需要扫描数据库，对数据库的IO压力很大。当频繁项的长度增加时，计算每个频繁项的时间会显著增加，算法的效率不高。

(2) GSP算法

Srikant等在Apriori-ALL 的基础上提出了GSP（Generalized Sequential Pattern） 算法。算法增加了时间的约束，定义了序列模式项与项之间的最小时间间隔，同时序列模式中的每一项没有必须来自同一事务的限制，挖掘的序列模式更加有层次。GSP的计算过程主要有以下4步：

（a）遍历序列数据库，产生序列模式（长度为1的序列模式）。

（b）候选序列集生成，利用长度为i的序列模式，产生候选的序列模式。在这过程中为了提高算法效率会进行连接和剪枝操作。连接指的是如果一个序列模式的第一项去掉后与另一条序列模式去掉最后一项后所产生的序列相同，那么将这两个序列模式进行连接，即将后者的最后一项添加到前者的最后，并删除后者。剪枝的具体操作是：如果一个候选序列模式的子序列不是序列模式，那么该候选模式不可能是序列模式，根据此规律，将其从候选序列中删除。

（c）通过遍历数据库，对每个选序列模式计算其支持度，产生序列模式，使用更新种子集；

（d）对（b）,（c）两步不断迭代，直到不在产生新的序列模式和候选序列模式的时候停止。

GSP算法与Apriori All算法相比有了很大的改进，数据转化过程中不需要事先计算频繁集。但是GSP 算法在还有以下不足：当序列数据库的规模很大时，生成候选序列的过程中会产生大量的候选序列。当序列模式比较长时，其对应的短序列模式规模也会很大。应用GSP算法时，依然需要对序列数据库进行频繁的扫描，数据库的查询时间开销很多。

（3）PrefixSpan算法

针对采用Apriori思想的序列模式挖掘算法在对规模较大的序列数据库进行挖掘时所面临的效率问题，研究者们提出了一种PrefixSpan算法。PrefixSpan的基本思想是采用分治策略，利用频繁序列模式作为前缀划分投影序列数据库，在划分的投影数据库中不断的进行迭代寻找序列模式。算法的主要过程为：

（a）扫描一次序列数据库，找出所有长度为1的频繁子序列。

（b）根据第一步中的所有频繁子序列，构造n个不同前缀，根据前缀将序列数据库划分为不同的投影数据库。

（c）在投影数据库中递归的挖掘频繁序列的子序列，当在无法产生新的频繁序列模式的时候停止。

和其他序列模式挖掘算法相比，PrefixSpan算法在计算过程中并没有生成大量无用的候选序列模式。通过前缀将序列数据库进行投影，这样可以在数据量不断缩小的投影数据库中进行序列模式搜寻，这很大程度上减少了数据的搜索空间，算法性能有了很大的提高。

2.4.3 分类分析

在数据挖掘中，分类分析（Classification Analysis）扮演了重要的角色。分类，是指将待分类的数据按照某种规律划分到不同类别中的过程。分类的目的是通过训练集训练出一个分类器或者分类模型，然后通过它将数据集进行划分。分类技术解决问题的一般过程如图2-2所示。

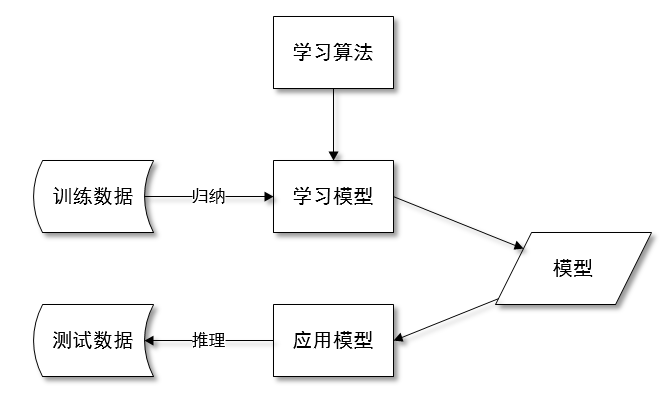


图2-2 分类分析一般过程

首先通过训练数据集对模型进行训练，模型利用算法寻找记录与已知类型标号之间的关联。然后根据训练出的模型形成应用模型对实际的数据集进行分类。整个分类分析的核心是分类器的构造。在利用规则对分类分析的分类器进行构造过程中不需要复杂的手段，方法简单易理解，对于基于神经网络的分类器构造方法则会产生很难理解的分类模型。

分类模型的评价指标一般是利用混淆矩阵来衡量，一般有准确率和错误率两个指标。

 （2-3）

 （2-4）

数据分类算法有许多种，常用的主要有：Ripper、ID3、C4.5、朴素贝叶斯等。

2.4.4 聚类分析

聚类分析（Clustering Analysis）和分类分析类似，但是又有很大的不同。聚类分析时，数据集的分类类型是未知的。聚类分析的本质是利用数据之间的相似性和相异性，对数据集自动划分成多个类簇，使得分类后的数据满足一定的约束条件。聚类的目的是，使得划分到同一个类簇中元素之间的相似性尽可能大，不属于同一个类簇之间的相异性尽可能大。相似或者相异的度量一般是基于数据元素之间的属性来确定。对于数值属性，常用的计算方式是对象属性间的欧式距离。聚类不需要进行模型的训练，对于聚类过程都是实时计算。在分类分析中如果模型建立起来后，在应用模型时的计算速度一般非常快，但是聚类分析不同，聚类的实时计算导致了其计算速度相对较慢。

聚类算法主要有以下几类：

（1）基于划分方法

对于给定的一个含有n个数据对象的数据集D，设定将要划分的子集的数目k(k<=n)，划分方法根据一定的算法将所有的数据对象分类成k个数据子集。这些子集满足以下条件（1）每个子集至少要包含一个对象；（2）一个对象只能属于某一个组（使用相似性描述对象属于哪个组是模糊聚类的思想）。划分时的目标一般是某个函数，常用的目标函数是平方误差，公式如下：

 （2-5）

E 表示数据集中的对象的平方误差和，p 代表了数据集中的某一点，是 分类的质心。

（2）基于密度方法

大多数聚类方法是根据对象之间的属性对所计算对象间的距离来进行聚类的，常用的距离是欧式距离。这种方法对于聚类的结果在几何空间上来说往往是圆形或者球形。很难发现任意形状的聚类。基于密度的聚类方法是对分类不断的增加新的对象，直到聚类的密度小于一定的设定参数。这种方式没有聚类结果形状的约束，可以适合多种聚类情景，发现任意形状的聚类。DBSCAN是常用的基于密度的聚类算法之一。

（3）基于层次方法

基于层次的聚类方法最后的聚类结果是将所有的数据构造成一个类似于树的聚类树。根据聚类是自下而上还是自上而下，层次聚类可以分为凝聚和分裂两种类型。自下而上的算法是从每一个单独的数据元素开始根据相似性计算规则，逐步的将所有的对象慢慢合并，最后达到某种条件或者合并为一类时停止，自上而下则相反。BIRCH是一个应用比较广泛的层次聚类算法。

（4）基于网格的方法：基于网格的方法相比于其他方法来说，它最大优点是在进行聚类时，整个聚类算法的运行时间与数据对象的个数没有关系，只与聚类时划分的网格数量有关系。首先网格聚类会把聚类的数据集分割成一定数量的网格单元，然后基于这种网格单元对数据进行聚类。采用这种方法的聚类算法在计算效率上会有一定优势。STING(Statistical Information Grid)是常用的基于网格计算方法之一。

2.5安全日志收集与存储技术

2.5.1 基于HDFS的文件存储技术

HDFS是Hadoop平台中的分布式数据持久化存储模块，处于整个Hadoop平台的最底层，在其上面一层为MapReduce[24]。在设计时，HDFS文件存储技术面向的是对需要存储的文件进行一次写入，然后可以多次读取的需求应用场景。其整个系统设计架构如图2-3所示。

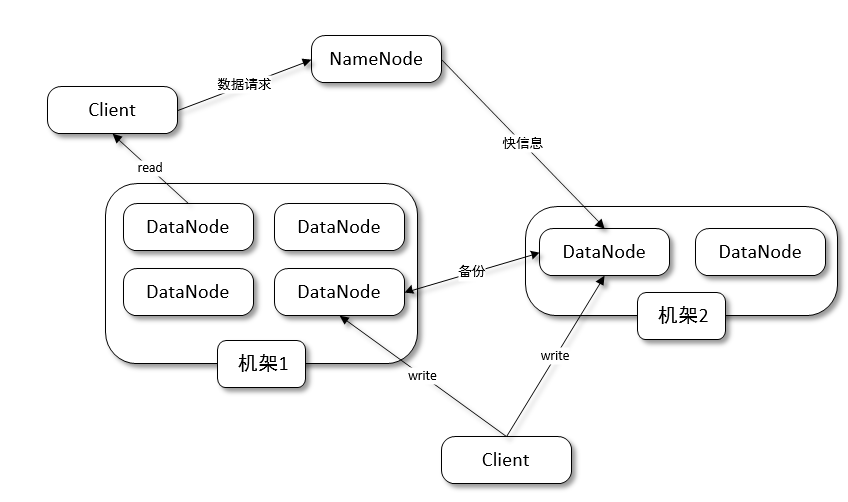


图2-3 HDFS文件系统架构

如图2-1在HDFS系统集群中，主要的组成节点分为两种：一种是管理节点（NameNode），一种是数据节点（DataNode），两者共同配合完成整个HDFS集群的数据存储。

NameNode是整个HDFS系统的管理节点，它存储着HDFS系统的命名空间。NameNode还需要对外部客户端（Client）访问集群中数据文件的行为进行有效的控制。在NameNode中的命名空间中保存了数据块到其所归属的具体文件的映射和文件的属性等，其维护了文件系统的结构树以及树内所包含的所有目录和文件。这些文件主要保存在FsImage的本地磁盘文件中。在NameNode中还保存了一个Editslog文件，它记录了整个HDFS集群中对文件的各种操作。例如，在HDFS中删除一个文件，就会在编辑日志文件中增加一条反映此操作的记录。在NameNode中为了解决分布式文件系统在多个数据同时读写时所产生的数据同步问题，NameNode中的编辑日志文件的设计是唯一的。这种设计也让系统的复杂性大大降低。

DataNode是HDFS的数据存储节点。它将数据以文件的形式存储在系统中，为了提高本地文件系统的效率，DataNode使用试探性的算法来确定每个DataNode本地磁盘中每个目录下的文件数目。如图2-1所示，HDFS使用块序列（Blocks）来对数据进行存储，HDFS会将大的文件切分成若干小块，然后分布式的存储在各个DataNode中。分块的大小一般情况下为64M[25]。DataNode的分块大小可以在创建客户机的时候人工设定。分块的大小要综合考虑各种因素来进行统一确定。如果分块过大数据块的传输时间就越大，由于HDFS中的任务计算是以块为单位在本地进行的，分块过大会造成参与计算的节点数量远少于集群节点数，不能充分发挥集群的计算效率。当分块过小则会有大量的时间耗费在数据块的寻址上。HDFS集群中的通信采用的是TCP/IP协议，NameNode只能被动的接收DataNode的RFC反馈。

Client给使用者提供的是一个操作系统界面接口，用户不用了解整个HDFS系统内部的实现方式或者集群架构就能对整个存储数据进行直接交互。HDFS的搭建设备没有很高的要求，普通的设备便可搭建起来性能较高的分布式文件存储系统。为了应对庞大集群出现宕机概率高的问题，HDFS采用了副本机制和心跳检测机制，文件的分块都是多副本存储在不同的DataNode上，当心跳检测发现某台机器出现问题后会自动在其他机器上进行自动的副本再备份，这个过程用户基本感觉不到，因此HDFS文件存储系统有很强的可靠性和容错性[26]。

安全日志数据集存储到HDFS文件系统中后，可以进行持久化存储。在HDFS设计之初，HDFS关注的重点在于整个数据系统的延迟，对于某一条记录的检索并没有过多的关注。HDFS设计的目的是为了解决海量数据的存储问题，并不是面向数据的快速查询读取[27]。因此，利用HDFS的分布式存储可以对收集到的海量安全日志进行持久化保存，便于以后的跟踪统计，但由于其查询和访问速度慢的缺点，对于安全日志的快速访问和查询分析需要利用ELK技术来实现。

2.5.2 Logstash日志采集技术

Logstash是一个诞生于2009年的一个开源日志采集项目。Logstash的主要设计功能包括三个：数据输入（Collect）、数据加工(Enrich)、数据输出（Transport）。Logstash几乎能够兼容所有的日志格式，功能十分强大。根据Logstash的设计功能，Logstash的总体架构如图2-4所示。

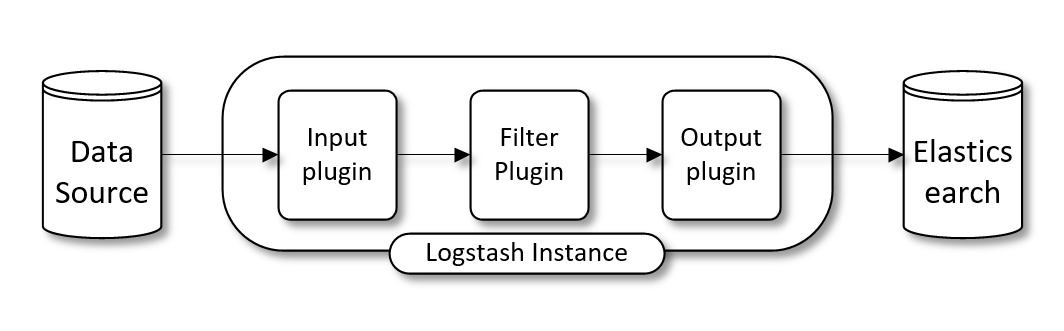


图2-4 Logstash模块架构

在解析日志时主要使用了Logstash的Input组件，它利用Multiline插件对日志进行封装。Input可以接受的来源很多，包括 Syslog消息传递以及JMX等。Filter中含有大量的日志解析过滤插件如Grok、Mutate、Drop等，使用这些插件可以对Input模块收集来的日志进行统一的格式化处理，删除无用的格式，补齐缺失属性、增加新的属性等。也可以通过Codec插件来简化日志解析的整个处理过程。在经过Filter模块后会生成标准的Json格式的数据。输出模块是Output组件。与Input组件一样，Output组件的输出方向很多，例如Redis、ElasticSearch、TCP、Email等。根据Logstash的工作方式不同分为shipper、Indexer。shipper主要负责对安全日志的收集，Indexer负责对安全日志进行解析。

2.5.3 ElasticSearch日志检索技术

ElasticSearch[28]是Elastic公司发布的基于Lucene的一款分布式搜索引擎。由于ElasticSearch是作为Apache许可条件下的开放源码发布的并且使用Java开发，因此，发展迅速，很快成为了目前企业级中广受欢迎的日志搜索引擎。越来越多的企业使用ElasticSearch来管理海量的日志文件。

Lucene是当前搜索领域一款性能优越、功能齐全的搜索引擎框架，但其具体功能的实现需要大量的Java语言开发工作 [29]。ElasticSearch基于Lucene框架进行功能封装，然后对外提供Restful风格的访问API[30]。使用ElasticSearch的用户在不需要了解Lucene底层的复杂实现原理就可以使用ElasticSearch提供的查询语言来进行数据查询功能。这大大减少了用户的学习成本和开发维护成本，提高了系统整体的研发效率。

ElasticSearch和传统的数据库不同，采用的Json结构的数据存储方式，是非结构化的数据，数据存储结构灵活、查询效率高[31]。ElasticSearch拥有很好的扩展性，在整个系统中ElasticSearch可以自动完成系统集群的负载均衡和数据分片备份，为用户提供稳定高效的服务。ElasticSearch基本架构如图2-5所示。

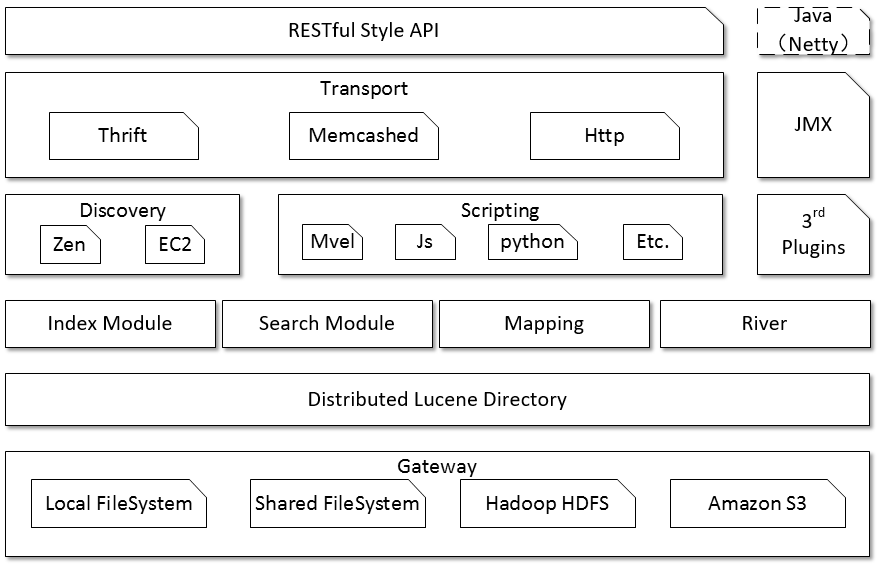


图2-5 ElasticSearch架构

Transport是ElasticSearch的交互层，该层描述了ElasticSearch的集群主机之间以及集群与客户端之间的交互方式，主要包括Http、Thrift、Memcached等多种协议。多种协议的提供使得ElasticSearch拥有很好的兼容性。

Discovery是ElasticSearch的发现模块，使用的发现机制有Zen和EC2两种。通过Zen机制，ElasticSearch可以自动的发现新加入集群的主机，过程分为两步：首先发送广播信息对存在的节点进行寻找，然后在使用多播协议进行主机节点间的通信。自动增删节点的工作机制，使得集群管理员很方便的对整个ElasticSearch集群进行管理，减少了人工配置工作量。如图2-5所示，ElasticSearch在进行功能扩展时支持的脚本种类很多，如：JS、Mvel、Python等，拥有很好的扩展性。

Index、Mapping、Search、River是ElasticSearch的功能模块，负责对索引的增、查、改、删（Create、Retrieve、Update、Delete, CRUD）等操作。Index、Search负责管理系统中的索引和数据的查询功能，在RESTful风格API中使用的主要命令有PUT、GET、DELET、POST。Mapping是ElasticSearch存储数据的模板，对于数据的类型可以使用默认的模板进行映射，也可以人工进行自定义动态模板（Dynamic Template）。River则是数据的读取功能实现，它能够帮助用户快速并行将数据读入集群。

Gateway是ElasticSearch的数据持久化模块。为了满足接近实时的数据搜索，ElasticSearch的索引是位于内存中的，当内存空间不够时，会将数据存入其他设备，比如Hadoop HDFS、Amazon S3等。Gateway使得ElasticSearch可以兼容这些持久化存储软件。为了满足海量日志检索存储，ElasticSearch一般采用基于HDFS文件存储系统来进行日志存储。

2.5.4 Kibana日志可视化分析技术

Kibana是基于B/S架构的安全日志可视化展示界面[32]。通过浏览器利用ElasticSearch的RESTful API就可以将集群里的数据进行可视化分析展示。主要展示类型包括折线图、柱状图等，使用者可以通过检索接口自定义展示视图。

Kibana在设计时参考了国外另一个非常优秀的大数据可视化分析软件Splunk，能够提供多样的日志展示形式。Kibana不仅能够进行可视化展示，还能够对存储在ElasticSearch中的日志进行简单的过滤、聚合分析。通过Kibana展示的日志数据必须包含时间戳字段，只有这样Kibana才能完成对日志的动态展示。得益于ElasticSearch的快速检索能力，Kibana可以达到秒级的动态展示。同时由于Kibana可以实现实时的可视化分析，因此也可以对日志的收集情况进行动态监测。Kibana配置简单，不需要复杂的数据查询聚类语句，在了解Aggregation接口的情况下就可完成对需要分析日志的可视化分析展示。如图2-6为定制的仪表盘动态监测日志收集情况。

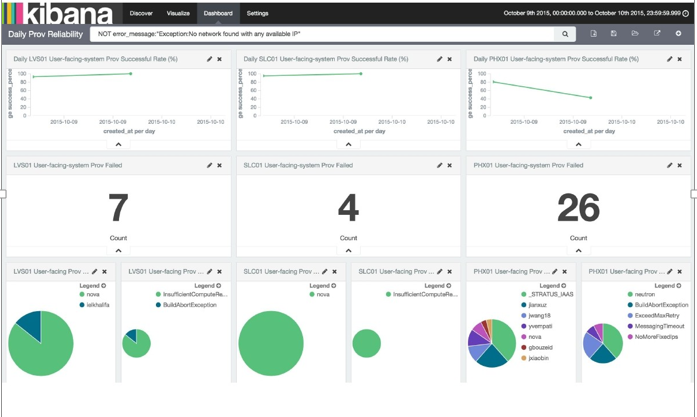


图2-6 Kibana Dashboard

2.5.5 Redis数据库技术

在分布式系统中，为了降低系统之间的耦合性往往需要增加一层消息队列，方便以后随着项目规模的变化来对系统集群进行扩展。消息队列是分布式系统集群的主机之间传递消息的缓存容器。消息队列本质上是一个队列，一般符合先进先出(First Input First Output，FIFO)的规则。队列有两种模式，生产者-消费者模式和发布-订阅模式，消息队列一般使用的是生产者-消费者模式。消息队列在对系统解耦的同时，能够保证数据传输的稳定性和安全性、兼容不同设备之间的数据处理效率差异。在日志收集分析领域常用的消息队列有Flume、RabbitMQ、Redis等。其中Redis由于其良好的性能被广泛应用于日志分析领域。

Redis是Pivotal公司支持，基于ANSIC语言开发的开源分布式数据库[33]。与Memcache缓存类似，其存储的数据一般情况下是缓存在系统的内存中。但是，Redis还支持数据的持久化功能，因此在提供高效率访问功能的同时保证了缓存数据的安全性。与传统数据库相比Redis的存储数据是非结构化的Key-Value形式，其存储的数据类型包括字符串（String）、链表（List）、集合（Set）、有序集合（Zset）、哈希类型（Hash）等[34]。在实现消息队列功能时一般使用双向链表数据结构（在数据量很少时会使用压缩列表）。

图2-7表示的是分布式情境下的生产者-消费者模式。在Redis中使用list来实现此模式[35]。生产者生产消息，将消息传递到消息队列中，消费者监听消息队列中的消息，当消息队列中存在消息时，消费者对消息进行消费。当生产者的生产速度大于消费速度时，消息队列可以对来不及消费的数据进行缓存，防止数据丢失。

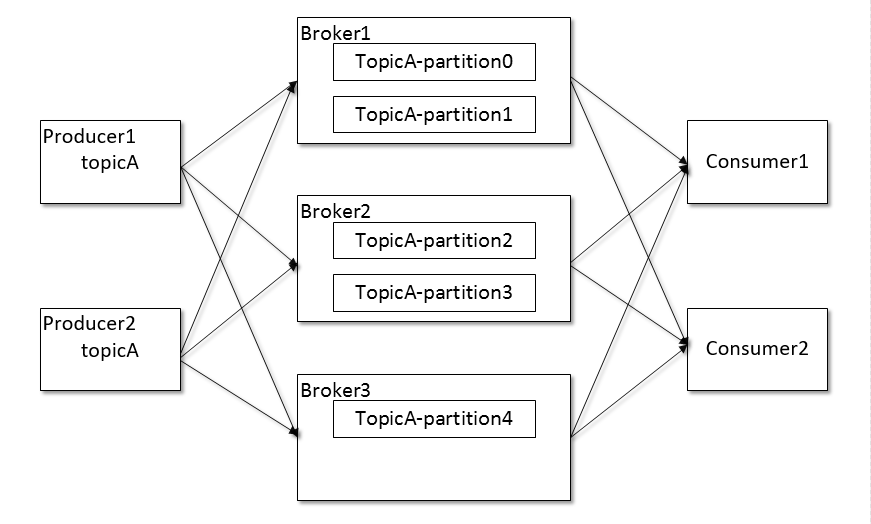


图2-7 生产者消费者模式

2.6 本章小结

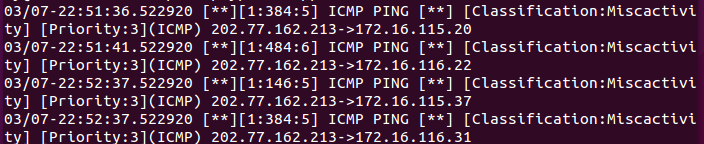
本章首先介绍了安全日志的相关概念、黑客攻击的一般步骤以及安全日志挖掘领域的相关知识技术。重点分析研究了数据挖掘过程中使用的相关算法，为后续的基于安全日志的攻击模式挖掘算法积累了理论知识。

第三章 基于模糊聚类分析和序列模式的攻击模式挖掘算法

直接利用数据挖掘中的关联算法对安全日志进行挖掘所产生的结果往往是场景混杂的，很难发现有意义的攻击模式知识。本章在对安全日志特点分析的基础上，提出了一种基于改进模糊聚类分析的安全日志聚类算法。算法能够根据安全日志属性间的相似性，对按照时间戳排序的全局安全日志进行准确、快速的聚类。然后使用基于序列模式的数据挖掘算法对聚类算法产生的攻击序列集进行序列模式挖掘，发现潜藏在数据中的黑客攻击模式信息。

3.1 安全日志特点分析

网络中多步攻击的攻击模式是由许多个不同的攻击步骤构成,攻击者通过一个步骤来达到一定的攻击目的，为下一阶段的攻击做准备。攻击者的攻击步骤所引发的安全日志信息是对入侵行为每个步骤的间接反映。因此，攻击者的攻击步骤之间行为的相关性与安全日志之间的相关性互为呼应[36]。如图3-1 是某入侵检测设备对某次攻击的记录。



（a）



（b）



（c）



（d）



（e）



（f）

图3-1 某次黑客攻击产生安全日志告警片段

这些告警日志片段反映的是对Soloris操作系统平台上Sadmind服务漏洞进行攻击的行为。攻击者对主机分别经过扫描探测、非法入侵、漏洞溢出获取权限来完成对主机的控制。然后，攻击者再利用这台被控制的主机作为媒介，继续对网络中的其他主机进行入侵攻击。

经过对安全日志的分析，安全日志的特点可以总结如下：

（1）从安全日志中的攻击类型属性方面看，日志的攻击类型属性反映了黑客多步攻击模式的多个不同步骤，黑客在进行这些攻击步骤的同时，安全设备会产生相应的告警类型，然后记录在安全日志中。这些攻击步骤往往是有策略性和目的性的，一个步骤的采用一般是为另一个攻击步骤做准备。

（2）IP地址上看，同一攻击场景下对同一网络的攻击所产生的安全日志之间，在源IP地址或者目的IP地址属性之间具有相似性。根据相似性可以一定程度上判断安全日志归属的攻击波次。

（3）安全日志属性类型上看，安全日志包括源IP地址、目的IP地址、源端口、网络协议类型、源端口、目的端口、产生时间等属性记录，属性的类型往往不是数值型的，而是分类或者字符串的形式。

（4）时间上看，一个攻击的攻击行为模式的发展是有一定时间顺序。攻击者在进行攻击时，会根据具体的情况采取的不同攻击步骤是有一定的时间顺序，不同的攻击步骤的时间顺序会反映在安全日志的记录中。另外，这些攻击步骤之间是存在一定的时间限制的，当超过某种时间，那么这些攻击步骤可能失效，例如，扫描到存活主机后，一定时间之后存活主机的工作状态可能发生改变。

（5）黑客的攻击步骤在安全日志中的反映是分散的。图3-1就是从大量的IDS告警日志中人工分析提取出来的，不同的攻击步骤之间穿插了大量的其他攻击事件告警信息。在大型的网络中，这种安全日志中对不同攻击所产生的场景混淆更加严重。人工很难从这些海量的日志中发现攻击者的攻击步骤。不同黑客，采取不同的攻击模式对于不同主机的攻击所产生的安全日志是同时存储在同一日志数据集中，如果直接对安全日志进行挖掘会产生场景混淆问题，如图3-2所示。A，B，C，D，E，F分别表示黑客对网站的攻击步骤所产生的日志。产生的日志按照时间顺序排序有多种可能情况，直接分析出黑客的攻击模式比较困难。一般情况下的黑客在进行攻击时会同时对多台目标进行攻击，同时相同的攻击行为模式也会重复出现在安全日志记录中。

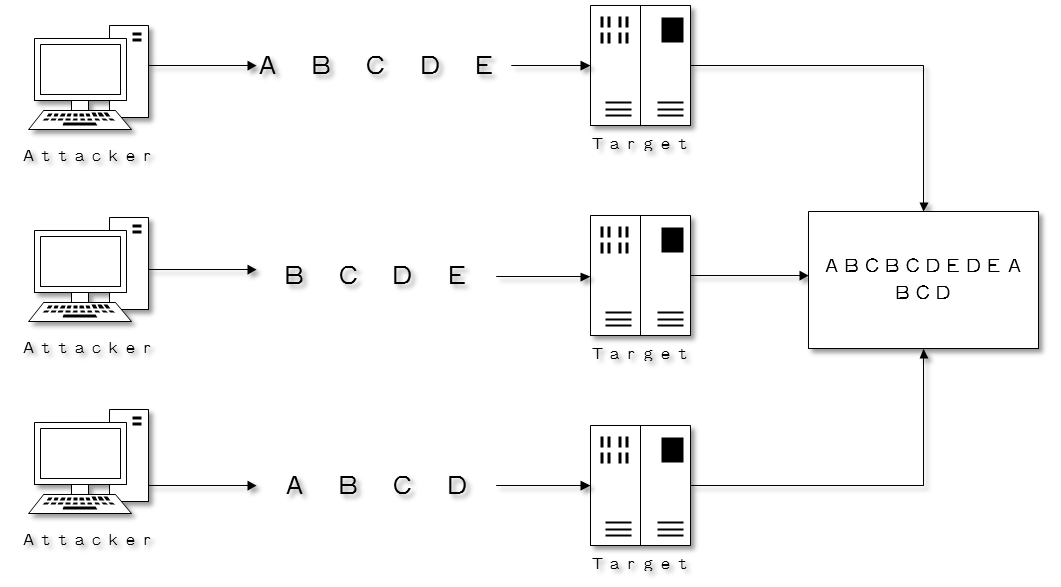


图3-2黑客攻击步骤和产生日志

3.2基于模糊聚类分析和序列模式的攻击模式挖掘算法流程

通过上一节对安全日志的特点分析发现，安全日志中隐含了构成多步攻击的攻击发生序列模式，因此可以使用序列模式挖掘算法通过对频繁序列挖掘的方式对隐含的攻击模式进行挖掘。但是，序列模式挖掘算法的数据集是交易事务数据库，而安全日志数据库不同于交易数据库。在安全日志数据库中，每一条安全日志是一个事件的记录，它并不是以序列集的形式存放在数据库中。根据上一节分析，同一攻击模式在日志属性间有一定的相似性，可以利用这种日志间属性的相似性将安全日志进行聚类，生成类似交易数据库的攻击序列集。在此基础上利用序列模式挖掘算法对攻击模式进行挖掘。算法的主要流程如图3-3。

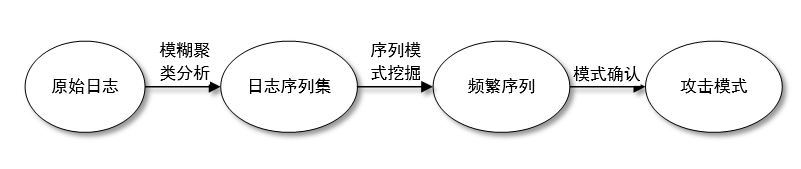


图3-3 算法主要流程

首先使用模糊聚类分析算法，根据日志属性的相似性，将按照日志产生时间排列的日志进行聚类分析，生成安全日志序列集；然后利用序列模式挖掘算法通过挖掘频繁集的形式对安全日志序列集进行挖掘；最后对挖掘结果进行分析，生成黑客的攻击模式。

如本文第一章研究现状对现有的攻击模式关联挖掘算法进行了讨论，对攻击模式研究方法中最具代表的有三种，基于因果关系的安全日志告警关联分析方法，基于机器学习的攻击模式发现方法，基于数据挖掘的攻击模式挖掘算法。基于这些方法在发现攻击模式方面存在诸多问题：在关联规则的定义过程中需要丰富的专家知识，因果关联规则的制定非常复杂，实现难度大；发现的往往是攻击模式片段，难以发现完整的攻击模式。

针对这些问题，本文首先利用同一攻击场景或者同一波次的攻击，所产生的安全日志间属性的相似性，将全局的安全日志进行聚类生成序列集。在此过程中使用了改进的模糊聚类分析算法，对日志的不同属性设置不同的权值更准确的描述日志间的相似性，同时对算法的计算效率进行改进。然后在模糊聚类算法的结果之上利用序列模式挖掘中的PrefixSpan算法挖掘出安全日志中隐含的攻击模式。本方法需要大量的人工专家经验，算法的挖掘结果所展示的攻击模式更加准确完整。

3.3 安全日志序列集生成

3.3.1 模糊聚类分析概述

传统的聚类分析算法在对数据集中的对象进行分类时，对类别元素的划分一般是非此即彼的二元分类方式，分类条件较为严格，在分类过程中的界限是分明的，属于硬划分的范畴[37]。但是在现实中，许多事物的分类类别比较模糊,并不能直接简单的对对象的类别进行划分。模糊聚类分析算法是根据Zadeh的模糊集理论提出的一种软划分的聚类方式。它是通过分别计算数据集中两个对象之间的相似度来进行分类，很好的解决了模糊边界问题，能够比较客观的反映真实世界的分类情况。由于在计算对象相似度的时候并不需要对象的属性为数值型，因此对于安全日志聚类来说模糊聚类分析比较适合。跟FCM等模糊聚类算法不同，模糊聚类分析算法在开始分类时不需要指定分类的数量，这也符合安全日志中攻击场景和攻击批次数量无法确定的现实情况[38]。基于以上分析，本文使用模糊聚类分析算法为基础对安全日志进行聚类来生成攻击序列集。整个流程如图3-4。

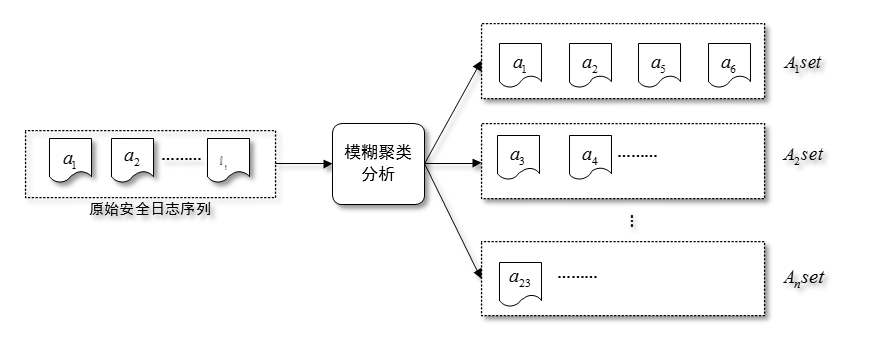


图3-4 攻击序列集生成流程

设数据集为为个待分类的对象样本。X中的每个样本点用一个维向量表示。表示第个样本有个属性。则样本的第个属性为。那么建立在上的模糊关系,的程度用相似度函数=表示。并且需要符合，。相似度函数的生成方式有：

（1）数量积法

 （3-1）

（2）夹角余弦法：

 （3-2）

（3）最大最小法：

  （3-3）

（4）主观评定法，根据属性实际意义主观确定。

确定了相似度函数后，则上的模糊关系矩阵可以用表示。一个模糊等价关系矩阵可以表示一个模糊聚类结果。对于一个确定的分类，可以通过确定了阀值的截关系矩阵来表示。传递闭包的方法就是使用了这种思想。使用传递闭包法进行模糊聚类分析聚类时，首先应该根据相似度函数生成模糊相似矩阵，然后使用平方法计算模糊矩阵的传递闭包，即分别计算直到出现时停止。然后根据动态阀值对传递闭包进行不同层次的聚类，形成不同的分类。利用平方法来计算传递闭包，进行模糊聚类分析的算法过程简单概括如下[39]：

（1）确定相似度函数；

（2）根据相似度函数建立相应的模糊相似矩阵；

（3）使用平方法对模糊关系进行计算，寻找到传递闭包时停止；

（4）根据不同的阀值进行分类，得到不同的动态聚类结果。

3.3.2 安全日志属性相似度函数

通过上一小节的研究可以看出，模糊聚类分析准确性的关键在于如何根据对象属性间的相似关系确定相似度函数。不同的数据集的属性特点不同，通过3.1节的分析可见，安全日志中的不同属性有其自己的特点，因此，需要根据其在实际情景中的具体含义确定每个属性的相似度函数。

定义（3-1）:安全日志警报事件，安全日志警报事件是一组五维向量， （Type是安全设备检测出的警报类型；Timestamp是安全日志产生的警报时间；SrcIP是源IP地址；DestIP是目的地IP；SrcPort是源端口）。

定义（3-2）：安全日志告警序列，，是一个按照时间戳排序的多维向量。

（1）总相似度函数

假设是安全日志告警序列中的任意两条日志,那么设定的总的相似度函数为日志中每个属性的相似度函数的加权求和：

 （3-4）

（2）端口相似度函数

常用的相似度函数的定义方法一般只适用于样本的属性类型是数值的情景下，而日志中的端口属性是非数值型属性。对于非数值型属性的度量，为了更好的描述端口间的相关信息，使用类似于层次树的相似度函数确定方法。如图3-5建立基本的端口层次树（这只是对端口初步的划分）对于具体的保护网络，根据端口功能可以生成更加复杂细致的多层树。

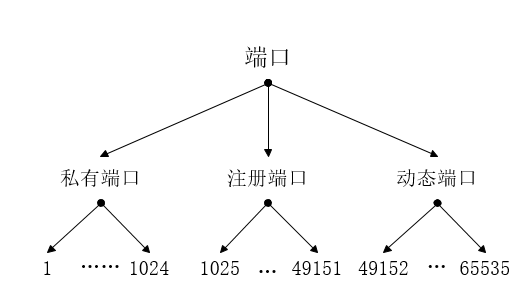


图3-5 端口层次树

两日志间的端口相似函数如下:

（3-5）

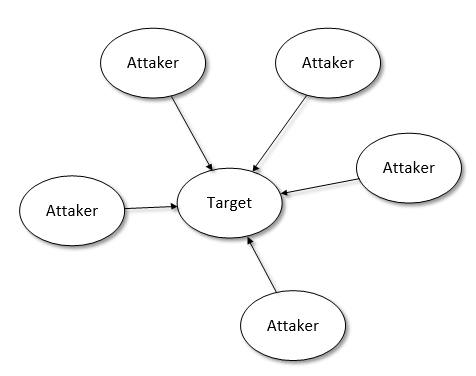
表示了两告警端口间的相似度，其中代表两个端口到达其最近公共祖先的最远距离，H为整个端口层次树的深度。

（3）IP相似度函数

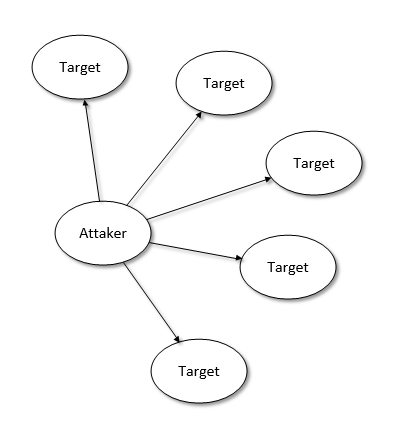
安全日志的IP属性有两个，一个是源IP地址（SrcIP），一个是目的IP（DestIP）。SrcIP反映了攻击行为的来源，源地址相同，那么这条日志很可能属于同一个攻击场景。SrcIP地址的相似性也反映了安全日志属于同一攻击行为的可能性。同样DestIP也时如此。SrcIP地址的相似度和DestIP地址的相似度的重要性在不同的攻击场景下会不一样的结果。在计算IP地址相似度时，两种IP地址需要统一综合考虑。

在安全日志属性中，IP地址既不是数值属性，也不是分类属性，两个IP地址间的相似性很难衡量。提出一种比较位数来计算IP地址间相似度的方法。是将32位的IP地址转换成二进制的形式时计算出来的从最高的数据位往较低的数据位进行比较统计出来的最多连续相等的位数。例如ip1 = 192.168.0.013也可写成11000000 10101000 00000000 00001011，ip2 = 192.168.0.211 也可以写成111000000 10101000 00000000 10001011。那么计算出来的。

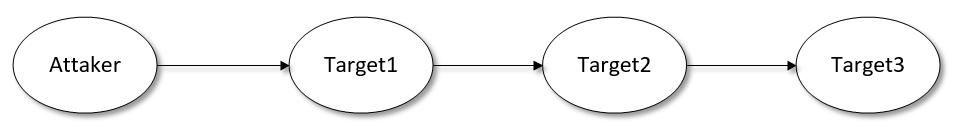
在安全日志中，IP地址包含源IP地址和目的IP地址，在不同的攻击模式情景下需要有不同的计算方法。通过对安全日志告警中的IP地址分析发现，攻击者攻击形式主要有以下三种方式，如图3-6。



（a）多IP攻击同一目标IP



（b）同一IP对多目标攻击



（c）跳板式攻击

图3-6 IP地址相关的不同攻击形式

为了能够准确的描述这三种情况下IP地址的相似度，综合考虑源IP地址和目的IP地址来对相似度进行计算。算法描述如下：

算法1：安全日志间IP相似度算法

输入：两个安全日志

输出：安全日志间的IP相似度

Step1:计算 的值。

Step2：如果或者的值较大，那么IP相似度为：

 （3-6）

Step3：其他情况下

（3-7）

Step4：输出

（4）时间相似度函数

假设，当它们之间的时间跨度大于某一个时间窗口时，可以认为两个安全日志告警事件属于同一攻击场景的可能性为零。当时间跨度在时间窗口之内时，使用高斯分布来计算两个警报之间的时间属性的相似性。这样可以更加符合实际情况。属性相似度计算公式如下：



（3-8）

3.3.3 模糊聚类分析算法改进

根据上一小节的安全日志相似度计算方法，可以通过模糊聚类分析利用传递闭包法对安全日志进行分类。但是在使用过程中存在一个问题，安全日志的数量庞大，从3.4.1小节分析中可以看出，基于平方法计算传递闭包来进行模糊聚类分析的过程中需要计算模糊相似矩阵，并且计算模糊传递闭包之后才能得到相应的模糊聚类结果。然后根据设定的阀值，计算相似模糊关系截矩阵。在这过程中计算消耗最大的是n维矩阵的乘法运算，计算时间复杂度为，那么基于平方法计算传递闭包来进行模糊聚类分析的总时间复杂度为，因此模糊聚类分析算法在应对大数据量的安全日志数据集时面临着时间效率问题。在计算过程中需要大量的内存和CPU计算资源。石陆魁等[40]提出了一种基于相似关系的模糊聚类算法。文献中使用数学理论证明了模糊关系的传递闭包的截关系与模糊关系的截关系的传递闭包相等，即：

 （3-9）

根据公式3-9就可以直接得到在设定阀值为时的具体的聚类结果。该算法时间复杂度为，空间复杂度为这极大的提高了模糊聚类分析的计算效率。基于此算法来生成安全日志攻击序列集的算法描述如下:

算法2：攻击序列集生成算法

输入：安全日志序列A，安全日志数量n，置信度水平

输出：安全日志攻击序列集S

Step1：将安全日志数据集A按照时间排序

Step2：令i=1，将添加到子类中，。

Step3：i++;

Step4：如果有k个子类，按照如下规则聚类：

（1）如果存在 满足，那么把添加到该子类最后一项。

（2）如果有两个元素 使得并且，那么合并子类和子类，并将元素重新按照时间排序，再把添加到新的子类的最后一项。调整修改后的子类编号。

（3）如果对任何一个子类，那么增加一个子类，。

Step5：判断i是否等于n，如果等于n则转到下一步，否则转到Step2。

Step6：去重，将每一类中连续出现的攻击类型合并为一个，输出安全日志攻击序列集S。

3.4 基于序列模式的频繁集挖掘

通过模糊聚类分析后将不同的场景下的攻击行为进行了聚类，形成了攻击序列集，为接下来的序列模式挖掘准备好了数据集。下面使用序列模式挖掘算法对上一节生成的攻击序列集进行数据挖掘，寻找出现频率大于最小支持度的频繁攻击序列,并以此为基础，经过人工分析之后形成能够完整反映黑客攻击行为的攻击模式。

在挖掘黑客的攻击行为模式时，关注的是黑客的攻击步骤，黑客的攻击步骤在安全日志中的反映便是安全日志中的攻击类型属性。因此，在此过程中只考虑告警类型这一类属性。同时，为了提高效率，在此阶段中将会把安全日志中的告警类型的字符串形式映射到它们所对应的数值上。由于数值型的数据只需要比较大小就可确定是否相等，因此可以节省大量数据寻找时间。从攻击序列集中挖掘攻击模式的算法流程如图3-7。

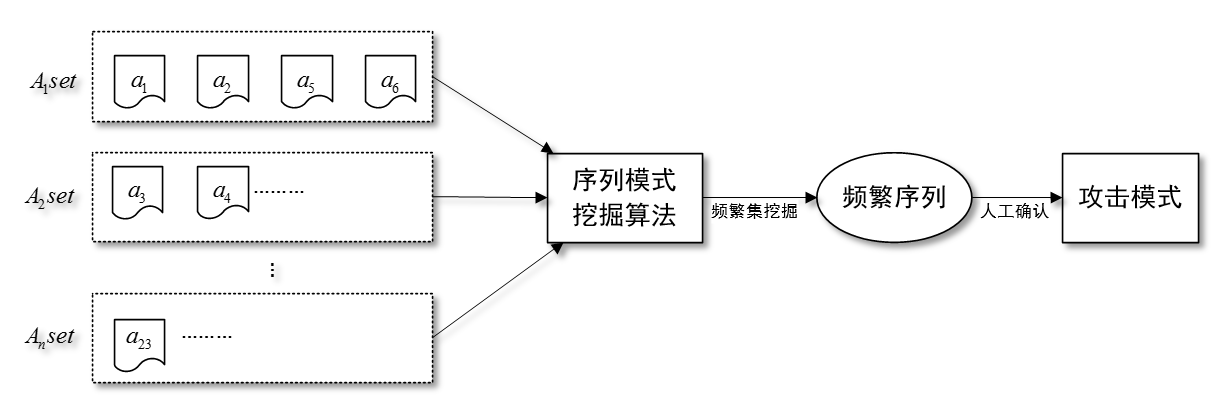


图3-7 攻击模式发现过程

正如第二章中所介绍的那样，序列模式挖掘算法主要三种：Apriori算法，GSP算法，和PrefixSpan算法。表3-1列出了三种算法的对比情况：

表 3-1序列模式挖掘算法特点

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 是否需要产生待选序列 | 数据存储方式 | 扫描空间变化 | 扫描需要次数 | 挖掘过程 |
| AprioriAll | 是 | 哈希 | 不变 | 结果的最长长度 | 循环过程 |
| GSP | 是 | 哈希 | 不变 | 结果的最长长度 | 循环过程 |
| PrefixSpan | 否 | 前缀树 | 缩小 | 2 | 递归过程 |

AprioriAll算法的计算中间过程，需要对可能的序列模式进行穷举，因此随着数据库规模的增大，数据项集数目的增加，产生的候选序列模式数量也急剧增长，因此AprioriAll算法需要占用大量的计算和存储空间，效率低下。虽然GSP算法和AprioriAll算法都是基于Apriori算法的思想而提出的算法，但是GSP算法由于使用了时间约束参数，因此在一定程度上减少了候选序列的数量，效率上与AprioriAll算法相比有了很大的提升。PrefixSpan算法在运行过程中，不会产生候选的序列模式，取而代之的是使用分治策略构建投影数据库，这极大的缩小了PrefixSpan算法的搜索空间，进而效率也有了很大的提升。因此对于大数据量的安全日志进行挖掘时，在保证内存足够的情况下使用PrefixSpan算法具有良好的效果。基于PrefixSpan算法挖掘频繁攻击序列的算法描述如下：

算法3：频繁攻击序列挖掘算法

输入：攻击序列数据库S；最小支持度mini\_support；最短频繁序列长度n

输出：频繁攻击序列模式

方法：通过调用SubPrefixSpan（,0,S）函数

SubPrefixSpan算法过程：

SubPrefixSpan（a, m, S|a）

参数意义：a是一个序列模式；m：a的长度；S|a：如果是空的那么是S否则是a基于序列集S的投影数据库。

Step1:扫描S|a，找到添加到a的最后一个元素中可以形成攻击序列模式的元素b。

Step2：对每个序列模式b，将b添加到a后形成序列模式c，如果c的序列模式长度大于n，则输出c。

Step3：对每个序列模式c，分别生成c的投影数据库S|c，然后递归调用函数SubPrefixSpan（c，m+1，S|c）

通过该算法之后，可以发现攻击序列集中的频繁序列。然而，并不是所有的频繁序列都是有意义的攻击模式，需要对结果进行分析后，才能得到有意义的攻击模式。本文算法的目的是提供一种方法，在减少所需的先验知识的情况下，从海量的安全日志中找到日志之间的关系，挖掘相关攻击模式知识库。通过该方法挖掘的攻击模式能够帮助入侵检测系统管理员制定更有针对性的入侵检测系统规则库。这在极大减轻了系统管理员负担的同时，也让能更好的了解黑客复杂多变的攻击手段。

3.5实验分析

3.5.1 实验数据集

本文采用的实验数据集是由美国国防部高级研究计划局（Defense Advanced Research Projects Agency，DARPA）赞助麻省理工学院（Massachusetts Institute of Technology，MIT）的林肯实验室，进行收集的入侵检测场景数据集DDOS1.0[41,42]。DARPA在2000年左右筹划资助了很多有关IDS系统的测试相关项目，目的是为了检验IDS的工作效率。DDOS1.0 数据集就是MIT林肯实验室在其支持下所进行的一项研究项目的产物。该数据集的采集方式是：搭建实验环境，然后寻找一些计算机安全相关研究人员对该实验环境中的网络进行攻击。通过部署在实验环境中关键节点的网络流量检测器，来采集相关的数据流量进而形成DDOS1.0数据集。

本次实验将使用DDOS1.0数据集为基础对本文的挖掘算法进行评估。首先采用Netpoke对数据包进行离线回放处理，共重复执行1天，利用RealSecure对回放流量进行入侵检测，收集RealSecure安全日志共得到673827条安全日志告警。表3-2展示了告警类型前五的数据。

表3-2安全日志告警类型分布（Top5）

|  |  |
| --- | --- |
| Type | Count |
| SCAN namp TCP | 561345 |
| ICMP Ping redirect host | 16359 |
| ICMP Ping | 6492 |
| RPC portmap request sadmind | 2434 |
| ICMP Echo Reply | 1253 |

3.5.2 实验环境

实验环境：Intel酷睿i7-4700MQ@2.7GHz 双核，8.00G内存，操作系统为Microsoft Windows 7，编程语言为Java。本文算法实现采用Java程序语言编程，利用My Eclipse编程软件实现相关算法的程序设计。My Eclipse拥有强大的开发测试集成环境，适合用于本文中的算法程序实现。

3.5.3 算法执行时间对比

A. f. Zhang等 [13]提出的基于数据挖掘的攻击模式挖掘算法（记为方法2），是目前安全日志攻击模式挖掘的算法中比较有代表意义的算法，和本文方法（记为方法1）一样，该方法不需要使用复杂的先验关联知识，也是从安全日志中挖掘攻击序列形成攻击模式。因此本文使用方法2与本文的方法进行实验对比。

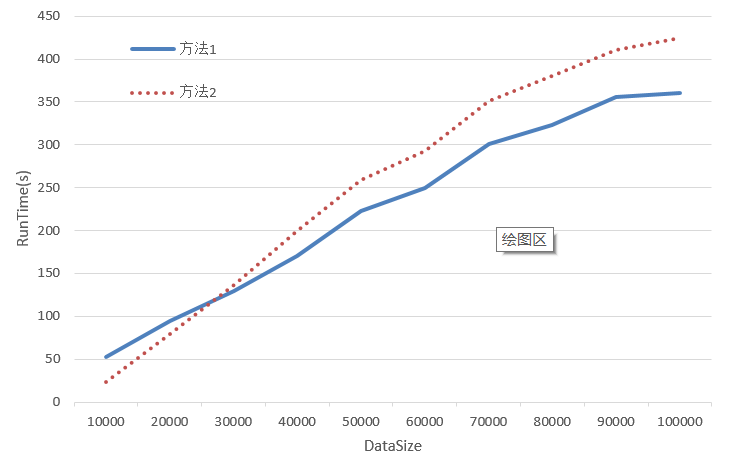


图3-8 算法时间对比

当数据集较小时方法1相比于方法2有一定的劣势。随着数据集的增大方法1和方法2的运行时间逐渐增加，但是方法1的运行时间逐渐小于方法2。造成这种结果的原因是：在序列集的生成方法上，方法2使用的是滑动窗口算法，该算法简单，仅仅只需计算时间属性，方法1使用的是改进的模糊聚类分析算法，在算法复杂性上相对较高。当数据集较小时，序列集的生成算法在时间上所占比例较大，因此在图像开始方法1算法运行时间大于算法2。但随着数据集的不断增大由于本文采用了PrefixSpan算法，该算法使用投影数据库进行优化，在序列模式挖掘阶段本文算法的耗时较少，因此随着数据集的增大本文算法在时间上的优越性便体现出来。由图3-6可见方法1在处理大数据集时有一定的时间优势。

3.5.4 算法结果对比

利用3.6.1小节采集到的数据集，通过方法1和方法2之后分别得到如图3-9和图3-10和的攻击频繁序列。根据图3-9的攻击序列结果通过分析可以得到如图3-11的攻击模式。图3-12是图3-10的图形展示。

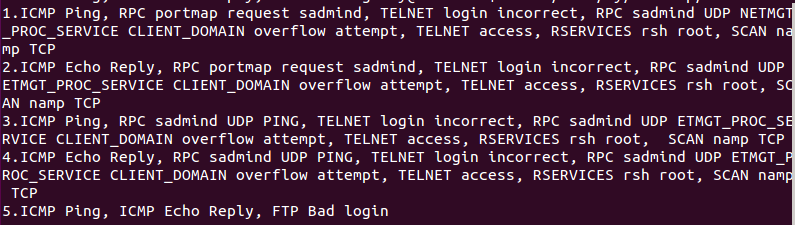


图3-9 方法1攻击序列结果

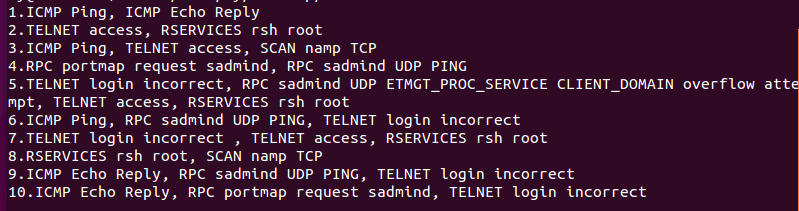


图3-10 方法2攻击序列结果

通过实验分析可知，方法1和方法2均在不需要先验知识的情况下成功地从数十万条安全告警日志中挖掘出了隐含其中的攻击模式，揭示了黑客攻击的不同阶段所采取不同步骤以及攻击者的攻击策略。揭示的攻击的具体步骤为：

（1）扫描网络。利用ICMP协议对网络存活主机进行探测。攻击者首先发送了基于ICMP协议的ICMP echo请求，然后对ICMP Reply的数据进行分析，以此确定主机的在线情况。

（2）判断在线的主机是否安装并运行进行攻击所要利用的服务或软件。在此过程中攻击者使用某种攻击软件工具发送伪造的request请求，类似第一步中的方式，对请求的回复进行监听来判断是否安装运行了sadmind服务。

（3）尝试登录。攻击者对在线的安装运行sadmind服务的主机进行telnet连接登录，结果失败。

（4）获取root权限。利用sadmind服务软件存在的漏洞，对安装该软件的主机使用缓冲区溢出的方式进行攻击来取得root权限。

（5）发动DDoS攻击。攻击者利用控制的主机对网络中最终的目标主机进行DDoS攻击。

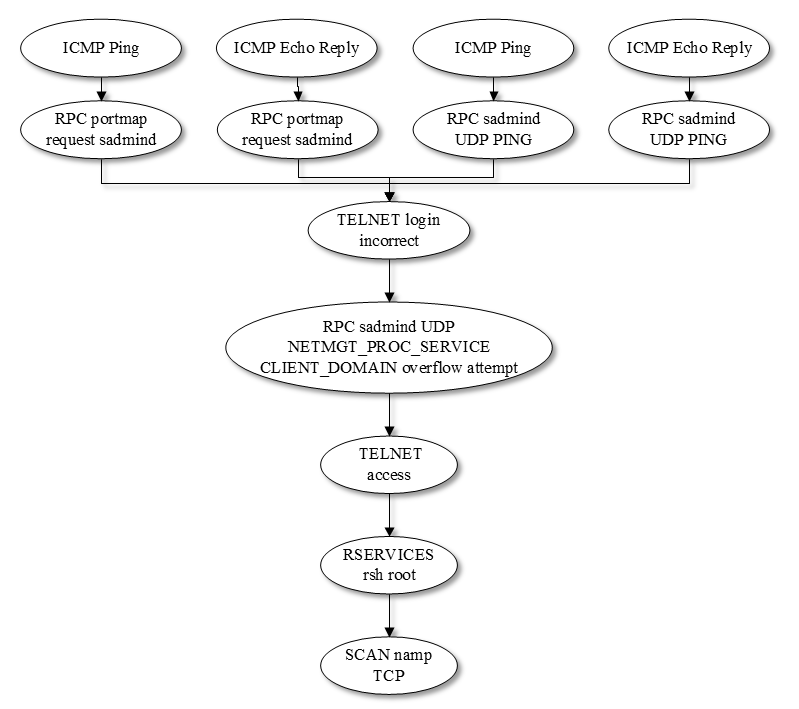


图3-11 方法1挖掘的攻击模式结果图

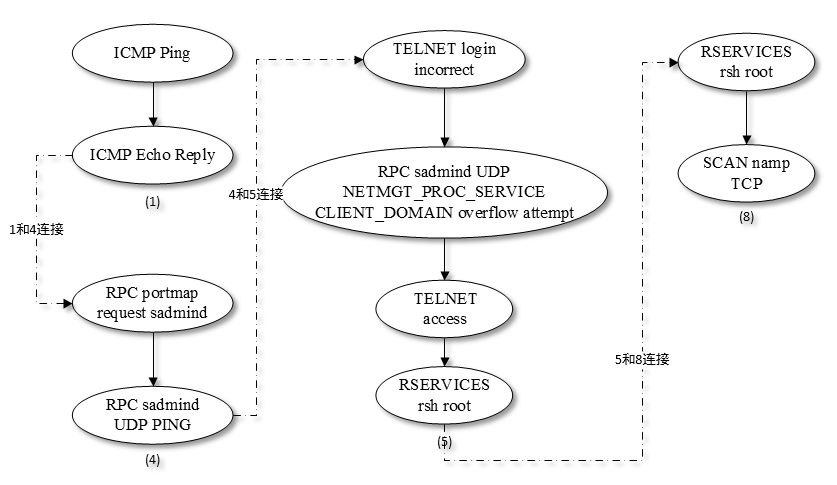


图3-12 方法2挖掘的攻击模式结果重构图

虽然方法1与方法2都挖掘出了隐含的攻击模式，但是从图3-10中可以看出，方法2的结果中存在很多没有意义的干扰项，经过重构分析之后才能发现攻击者的攻击模式。图3-12的攻击模式是利用是通过序列1,4,5,8连接而成。而方法1中这种干扰项很少。这是因为方法2中仅仅使用了固定的滑动窗口来生成攻击序列。这样很容易造成攻击模式的断裂，在后期分析中还要进行攻击场景的重构才能得到较好的结果。同时由于攻击场景的割裂便产生了很多短小的频繁攻击序列，这也是方法2中干扰结果数量大于方法1的原因。而方法1通过模糊聚类分析算法根据日志的属性将日志进行聚类之后再挖掘频繁攻击序列，这样能够减少攻击场景断裂的情况，进而帮助安全管理员更加高效的从序列结果分析攻击者的攻击模式。

3.6 本章小结

在本章中，主要介绍了基于改进模糊聚类分析和序列模式的攻击模式挖掘算法的流程。方法首先将安全日志数据转化为警报序列，并通过快速模糊聚类分析算法根据安全日志之间的相似度，将处于统一攻击场景下的序列聚集在一起转化为候选攻击序列集，再利用序列模式挖掘算法从候选攻击序列集中挖掘出多步攻击行为的发生序列模式。不同的攻击行为序列模式反映了黑客攻击的不同攻击步骤。最后对该算法进行了实验评估。

第四章 基于ELK技术的攻击模式挖掘系统

基于上一章提出的一种基于改进模糊聚类分析和序列模式的攻击模式挖掘算法，在本章中结合ELK技术，提出了一个以ELK技术为基础的攻击模式挖掘系统。在本章将对该系统的各个模块进行详细介绍。

4.1 系统概要与整体架构

攻击模式挖掘系统的主要目的是：对不同安全设备产生的多源安全日志信息进行收集，整理，进行统一的格式化存储，在此基础上利用相关的数据挖掘算法对检索存储的海量安全日志信息进行数据挖掘，分析日志中隐含的攻击模式知识，为防范网络攻击提供支持。为了满足需求，本文基于ELK技术设计了如图4-1所示架构的攻击模式挖掘系统。系统主要分为安全日志采集模块，安全日志处理模块，安全日志存储模块，攻击模式挖掘模块，系统监控模块。

安全日志采集模块主要是对产生安全日志的数据源的日志进行采集，定时主动的或者被动的采集安全设备的安全日志。采集的数据主要有两个流向，一个是直接存储，作为原始日志封存以备日后查验，另一个是发送到缓存模块等待后续处理。

安全日志处理模块是对日志进行格式化处理，将不同格式的日志进行统一的字段切分，形成统一的日志格式，并解析出每条日志的产生时间作为每条日志的时间戳。同时为了弥补日志解析和日志采集模块的速率不匹配问题，需要增加日志缓存模块，防止日志丢失的情况产生。

数据存储模块主要是对安全日志数据进行存储和满足攻击模式挖掘模块对安全日志的查询需求，为了满足需求采用ElasticSearch分布式日志检索技术和HDFS分布式文件存储技术。

攻击模式挖掘模块主要是利用本文第三章提出的基于改进模糊聚类分析和序列模式的攻击模式挖掘算法定时或者依照管理员设定主动的对安全日志进行分析处理，挖掘出隐含在海量安全日志中的攻击模式，并存入数据库。

系统监控模块主要是对整个平台的内存消耗，CPU使用率等指标进行监控。总体上把握系统的运行状态，保证系统的稳定性。

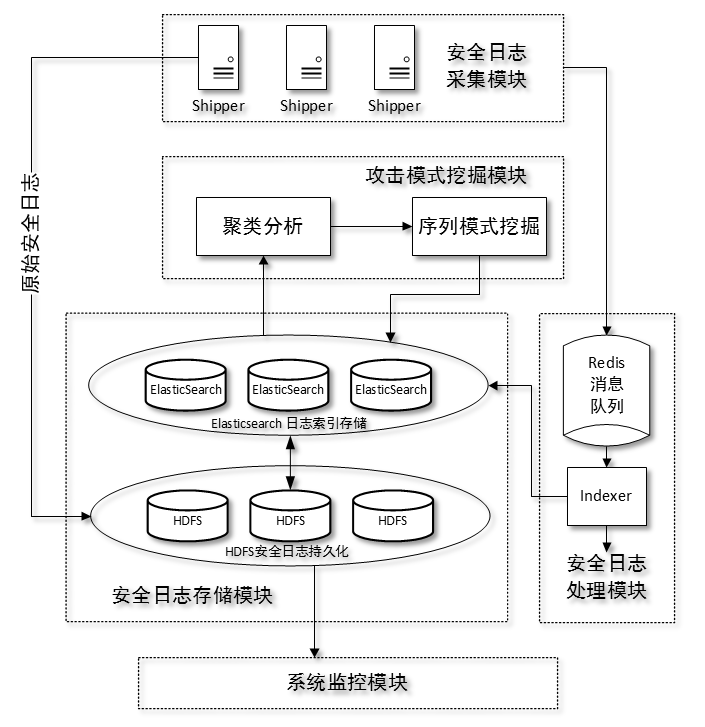


图4-1 基于ELK技术的攻击模式挖掘系统

4.2 主要功能模块

4.2.1 安全日志采集

为了满足分布式多源日志的收集需求，本文使用Logstash来完成日志的收集。Logstash根据在数据流中的不同角色分为三种：shipper、broker和indexer。在采集阶段Logstash主要担当shipper角色。根据采集方式不同Logstash有两种收集方式,一种是主动式的获取，作为安全日志产生服务器上的一个Agent模块来对日志进行采集。另一种是被动的接收发送安全日志。对Logstash的采集模块进行配置时主要使用了input和output插件，在shipper模式中input与output之间没有其他处理直接进行数据的转发。数据流向如图4-2。采集模块的具体的配置代码如图4-3所示。

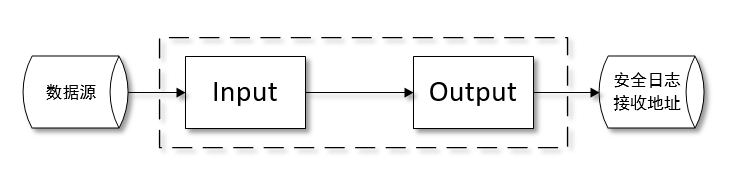


图4-2 shipper模式下的数据处理流程

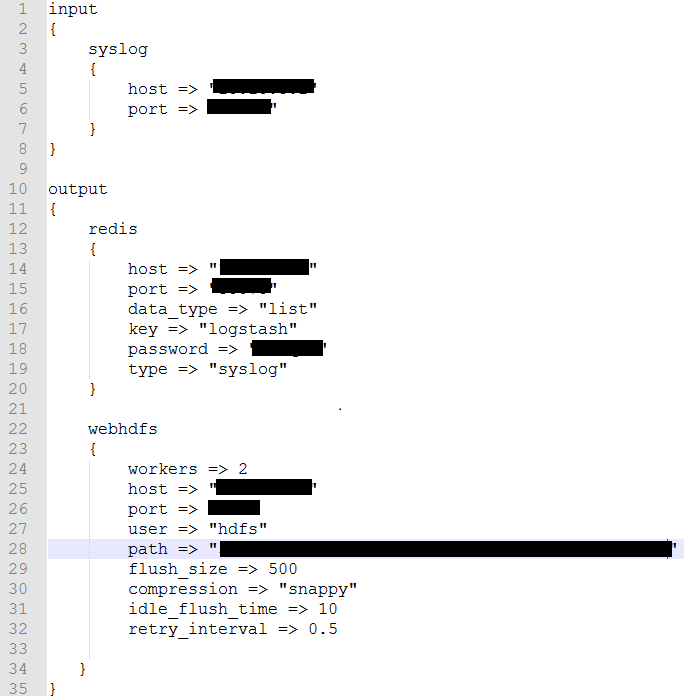


图4-3 采集模块Logstash的具体的配置代码

在采集模块中使用了syslog插件被动的采集安全设备产生的安全日志，可以根据设备发送或者产出安全日志的模式不同使用不同的配置参数，如文件、数据库等。Syslog配置模块中的host和port分别是安全设备的IP地址和端口。在Output插件部分使用了Logstash 软件自带的Redis配置插件和由Github上获取的Logstash-output-webhdfs插件。根据配置文件可以看出安全日志采集后有两个流向，一个直接作为备份直接存入HDFS中，另一个发送到安全日志处理模块的Redis缓存。从配置文件中的data\_type参数可以看到使用的是redis的list类型作为消息队列对日志进行缓存。整个安全日志采集阶段的数据流向如图4-4所示。

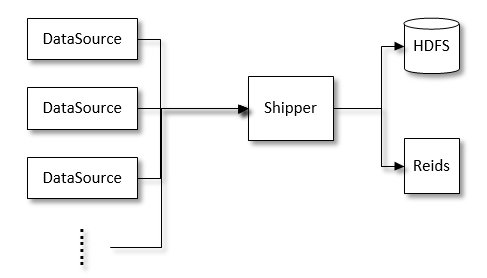


图4-4 安全日志采集阶段的数据流向

4.2.2 安全日志处理

日志处理模块是对日志进行格式化处理，将不同格式的日志进行统一的字段切分，形成统一的日志格式并为每一条安全日志设置时间戳。同时为了弥补日志解析和日志采集模块的速率不匹配问题，增加日志缓存模块，防止日志丢失。安全日志处理模块主要分为两个部分，分别是基于Redis的日志缓存部分和日志格式化Indexer部分。

（1）Redis缓存部分

缓存部分是将分布式数据库Redis作为消息队列来对采集模块收集来的安全日志进行缓存，然后提供给Indexer部分进行安全日志的解析处理。Redis-Logstash插件工作流程如图4-5所示。

由于日志来源的多源性和日志流量的突发性，安全日志采集模块日志的传输速度并不稳定。使用Redis消息队列可以防止在日志流量突发情况下日志丢失情况的发生。在上一节shipper的配置文件中，output模块的data\_type设置为了list属性，这使得Redis自动选择消息队列模式。

在Redis的消息队列模式中，消息队列的实现是由Redis的列表结构来实现。分布式系统中，一个队列被称为是一个主题。为了保证消息的同步和唯一性，消息队列中的某一条消息只能属于一个主题。在本系统中消息生产者是采集模块中Shipper模式下的Logstash软件，消费者是数据处理模块中的Indexer模式下的Logstash软件。消费者不断产生消息放入队列中，生产者对消息进行消费，消费完成后，消息就不会在消息队列中存在了。

初始时，Redis中并没有队列，通过如表4-1中的Redis操作参数完成对消息队列的实现。LPUSH/LPOP操作命令能够对Redis的list数据结构的表头进行操作，RPUSH/RPOP命令能够完成对list的表尾的操作。消费者在进行消息的消费时，会从队尾使用RPOP命令对消息队列中的消息进行消费。生产者则是使用LPUSH/RPUSH将消息压入消息队列中。当使用LPUSH命令压入消息时符合先进先出的队列模式，一个消息的生产时间顺序决定了它的消费顺序。但是，有一些特殊情况下存在某些比较重要的消息，那么生产者可以使用RPUSH命令将消息直接压入表尾，这使得该消息可以不用参加消息的排队，直接被消费者使用RPOP命令对消息进行消费处理。本系统中消息生产者和消息消费者操作Redis列表的方式如图4-6所示。

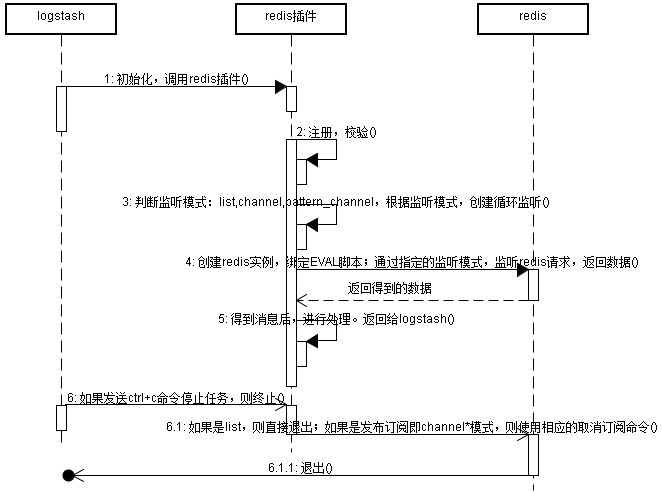
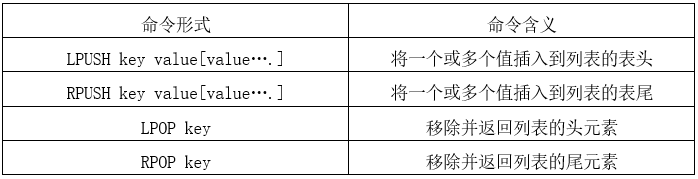


图4-5 Redis-Logstash插件的工作流程

表4-1 Redis列表相关常用命令



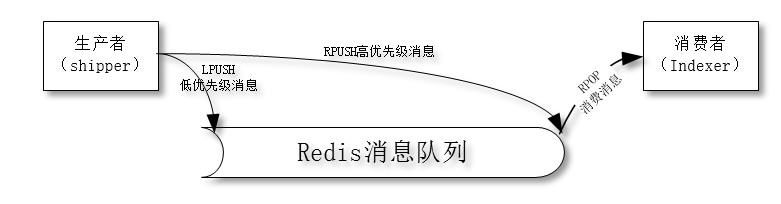


图4-6 Redis消息对列模式

（2）Indexer部分

Indexer部分使用的是Logstash的Indexer模式，在从Redis消费数据后，经过核心过滤插件Filter，对原始的安全日志信息进行过滤，解析字段，生成需要的形式，然后输出到ElasticSearch中。在Indexer模式下的数据流程如图4-7所示。

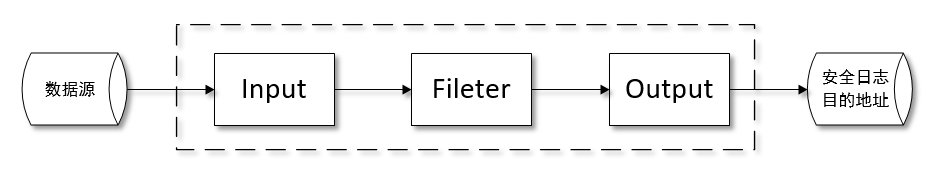


图4-7 Logstash在Indexer模式下的数据流向

在Filter中将原始多余的字段丢弃，只保留在攻击模式挖掘过程中需要的字段，主要包括，源IP地址（SrcIP），目的IP地址（DestIP），源端口号（SrcPort），日志产生时间（Timestamp），日志告警类型（Type）。保留字段如表4-2所示。

表4-2日志保留字段及类型

|  |  |  |
| --- | --- | --- |
| 字段 | 含义 | 类型 |
| SrcIP | 源IP地址 | IP地址 |
| DestIP | 目的IP地址 | IP地址 |
| SrcPort | 源端口号 | Int |
| Timestamp | 日志产生时间 | Data |
| Type | 日志告警类型 | String |

为了对多源的安全日志进行格式统一，需要针对不同的日志格式配置不同的Filter解析代码。图4-8展示了某设备的日志解析配置代码片段。在Grok部分使用正则表达式对日志的各个字段进行匹配，生成Key-Value数据格式的数据。然后使用drop来删除不必要的字段，这样可以减少攻击模式挖掘算法的数据搜索规模，提高数据查询效率。

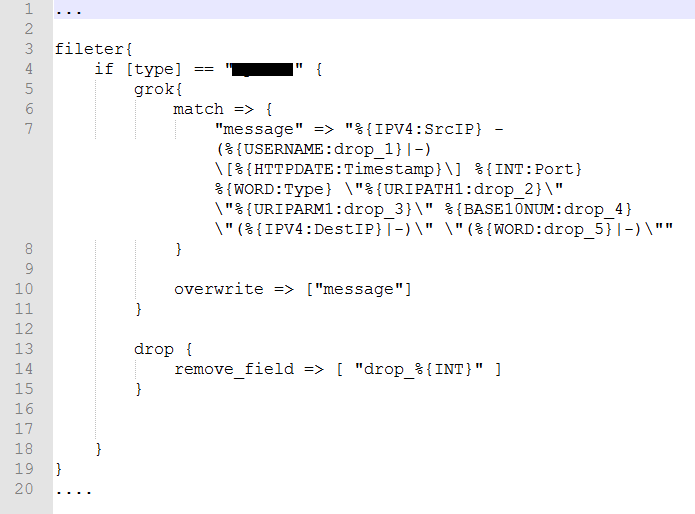


图4-8 某厂家IDS日志解析配置代码片段

4.2.3 安全日志存储

安全日志存储模块的主要目的有两个，一个是为攻击模式挖掘模块提供高效的海量日志查询速度，保证算法的执行效率；另一个是对采集的安全日志进行持久化存储，保证数据不被丢失。为了满足需求，本系统使用ElasticSearch分布式搜索引擎技术和HDFS分布式文件存储技术。整个日志存储模块的数据流入流出情况如图4-9所示。数据的流入主要包括原始安全日志，经过处理模块统一格式后的日志，攻击模式挖掘模块挖掘出的攻击模式结果。流出主要是监控信息，为攻击挖掘算法提供数据支持。在存储模块内部，ElasticSearch提供安全日志检索查询功能，Hadoop主要利用其HDFS分布式文件存储技术对安全日志进行持久化存储。二者之间交互通过ElasticSearch-Hadoop API进行数据传输。

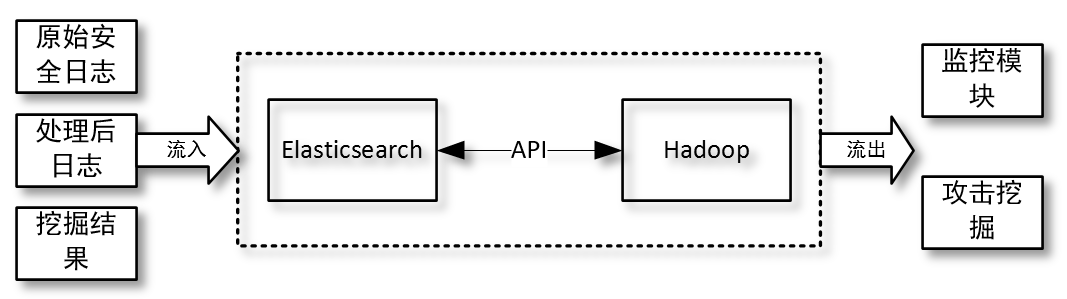


图4-9 日志存储模块数据流入流出情况

在存储模块中使用了动态模版（Dynamic Template）来对日志的类型进行动态匹配，这样可以让数据的格式根据数据内容不同进行动态调整，有利于提高存储模块的查询效率。如图4-10所示动态模版的部分代码片段。代码根据属性名称对Logstash传输的日志进行类型匹配。



图4-10 动态模版部分配置代码片段

4.2.4 攻击模式挖掘

攻击模式挖掘是系统的核心功能部分，主要是对系统收集的海量日志进行数据挖掘。该模块基于本文第三章提出的基于改进模糊聚类分析和序列模式的攻击模式挖掘算法对存储的安全日志进行数据挖掘，挖掘出安全日志中隐藏的攻击模式。主要流程如图4-11所示。

攻击模式挖掘的具体的步骤如下：

（1）开始，接到管理员或者定时程序对存储安全日志进行挖掘的指令。

（2）模糊聚类分析，根据聚类参数对安全日志进行聚类，生成攻击类似交易数据库的攻击序列集。

（3）序列模式挖掘，对模糊聚类分析产生的聚类结果使用PrefixSpan算法进行频繁集挖掘，输出频繁子序列。

（4）对上一步的挖掘结果进行人工分析，生成攻击模式，存入数据库。

（5）结束。

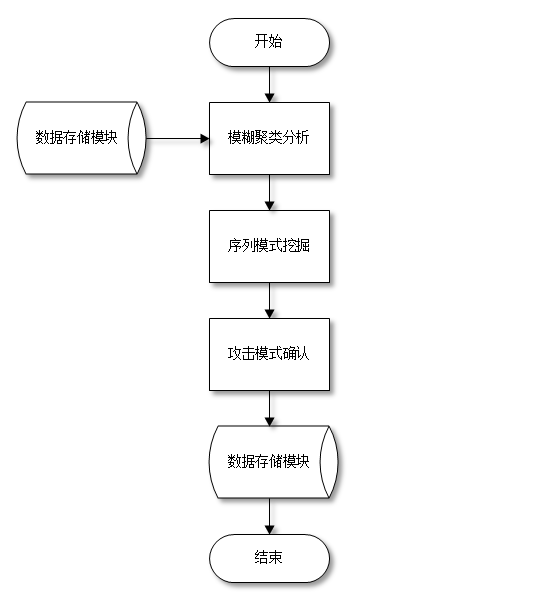
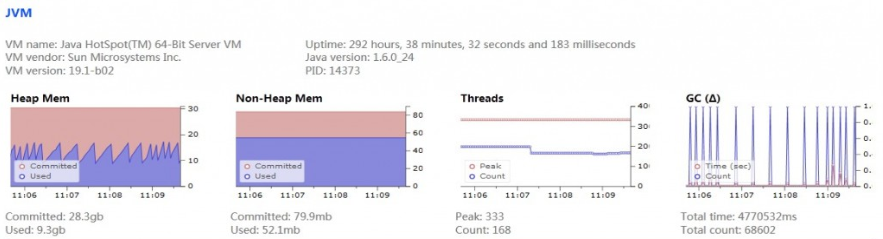


图4-11 攻击模式挖掘流程

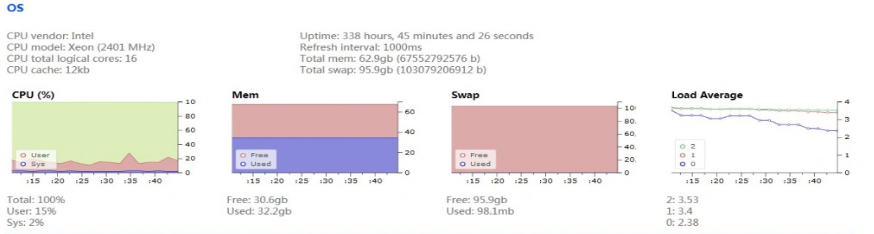
在攻击模式挖掘模块，由于使用的是ElasticSearch统一提供的访问、写入API，因此该模块的与其他模块的耦合性很低。对于存储在ElasticSearch和HDFS中的数据，我们可以根据分析算法的不同对该模块的算法进行更换，该模块具有良好的扩展性和兼容性。

4.2.5 系统监控

系统监控模块主要负责监控整个系统的运行情况，在大型的数据集群中，系统的状态监控模块十分必要，依靠人工对机器运行状态监控需要耗费很多精力。通过统一的系统监控运行模块，完成对整个系统集群的运行状态监控，能够让系统的运行更加稳定，及时发现异常情况。系统监控模块的展示界面如图4-12所示。在JVM监控模块主要反映了系统某台机器中的堆内存、线程数量、垃圾回收等实时现状。操作系统的监控主要包括CPU、内存、IO数据交换、以及磁盘存储情况进行分析。该模块的实现是利用Bigdesk插件实现。



(a) Java虚拟机状态（JVM）状态监控



(b) 某主机操作系统状态监控

图4-12 状态监控

4.3 本章小结

本章主要阐述了基于ELK技术的攻击模式挖掘系统的设计和搭建方式。首先，概括介绍了该系统的整个系统框架结构；然后，对安全日志采集、安全日志处理、安全日志存储、攻击模式挖掘、系统监控等模块的数据流程和处理逻辑进行了分析讨论。

第五章 实验与分析

在第四章中主要介绍了基于ELK技术的攻击模式挖掘系统的设计与实现方式，本章主要是搭建实验环境，采集真实的网络安全日志，并在此实验环境下利用真实数据集对攻击模式挖掘算法进行评估。

5.1 实验环境搭建

本章实验环境中核心集群的搭建主要基于实习单位的模拟仿真实训平台实现。如图5-1所示，在模拟仿真平台上配置了五台虚拟机，一台虚拟机配置Redis和Logstash的Indexer模式作为数据处理机，三台虚拟机ES01、ES02、ES03部署ElasticSearch搭建ElasticSearch分布式日志检索集群。还有一台Caculate虚拟机负责攻击模式的挖掘计算。另外还有一台服务器部署Hadoop，作为安全日志的持久化存储服务器。虚拟机和服务器的硬件配置如表5-1所示。软件名称及版本如表5-2所示。

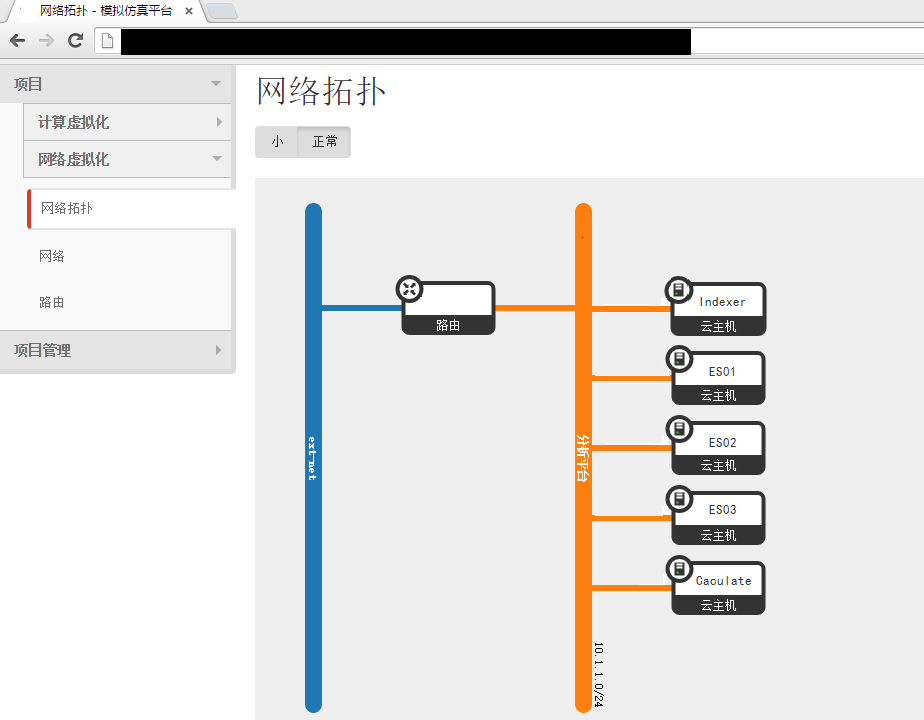


图5-1 核心集群网络拓扑

表5-1 实验环境主机硬件配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机名称 | CPU | 操作系统 | 内存 | 硬盘 |
| Indexer | 虚拟2核 | Ubuntu-14.10-64位 | 8G | 40G |
| ES01 | 虚拟4核 | Ubuntu-14.10-64位 | 8G | 80G |
| ES01 | 虚拟4核 | Ubuntu-14.10-64位 | 8G | 80G |
| ES01 | 虚拟4核 | Ubuntu-14.10-64位 | 8G | 80G |
| Caculate | 虚拟8核 | Ubuntu-14.10-64位 | 8G | 80G |
| HDFS服务器 | 4CPU | Ubuntu-14.10-64位 | 16G | 2T |

表5-2 实验环境所用软件名称和对应版本

|  |  |
| --- | --- |
| 软件名称 | 软件版本 |
| Java Development Kit (JDK), | 1.7.0\_80u |
| ElasticSearch | 2.4.3 |
| Logstash | 1.5.4 |
| Kibana | 4.1.2 |
| Redis | 2.8.19 |

5.2 实验过程

5.2.1 安全日志收集测试

此次实验过程的安全日志数据源主要来自于某实习单位的IDS告警日志。在IDS服务器上部署Logstash使用Shipper模式对安全日志进行数据采集。数据源是多个监控不同网络的不同厂家的IDS设备，采集的日志是多源异构的，比较符合大型网络的安全日志产生情况。根据实验环境和数据源构建了如图5-2所示的网络拓扑。

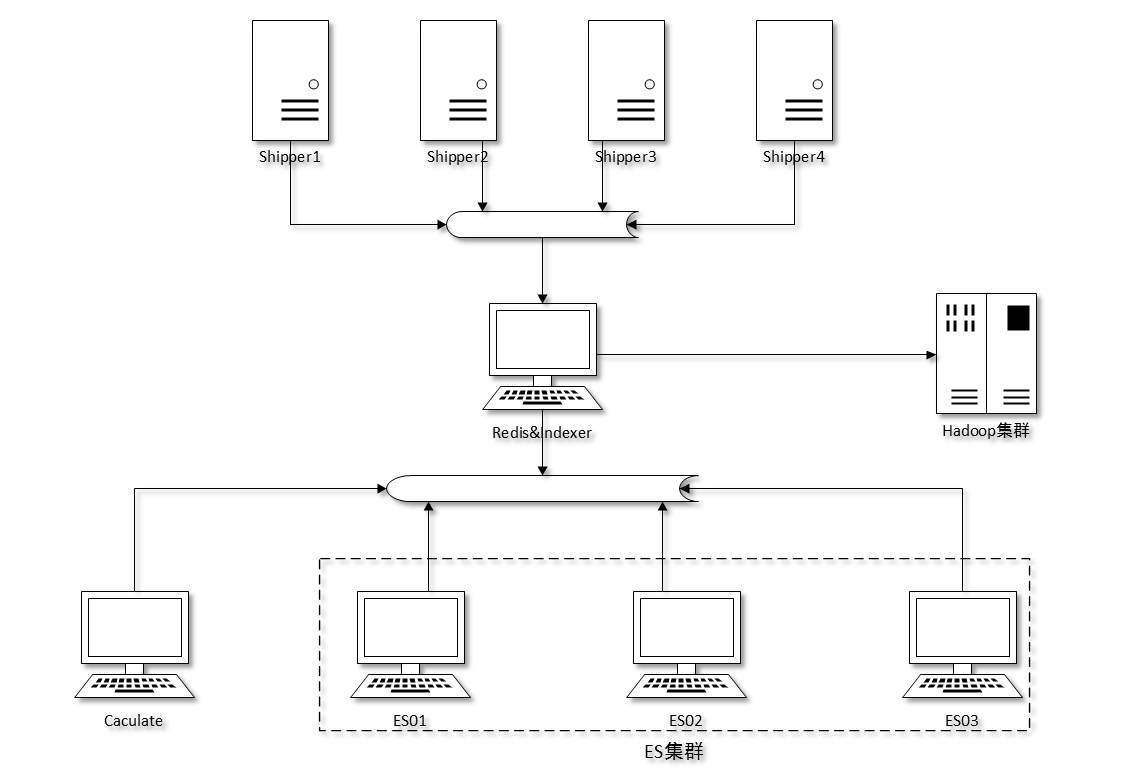
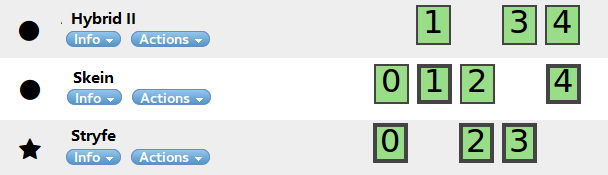
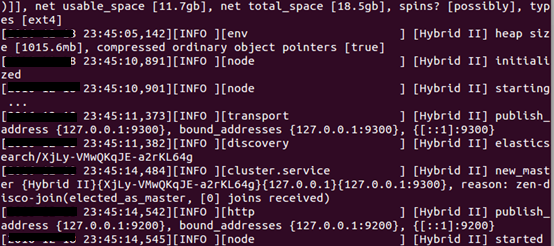


图5-2 网络拓扑

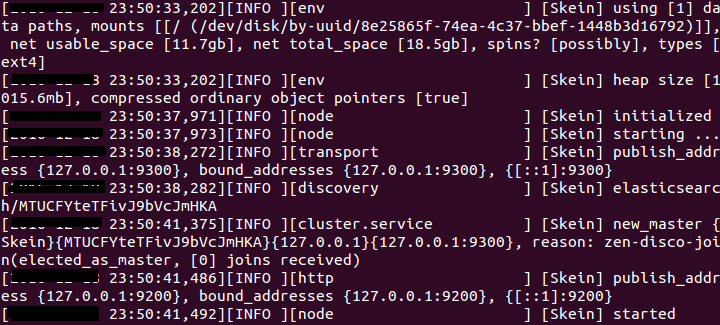
从图5-3可以看出ElasticSearch集群中的三台主机中的ElasticSearch线程的启动情况，同时三台主机在集群的中名字分别为Hybrid II、Skein和Stryfe。集群中三台主机通过自主选举，选举出ES03（Stryfe）为Master节点，其余两台为Slave节点。同时集群采用了分片机制，分片数量为默认值5。图5-3（a）中展示了集群分片的分布情况。由于分片备份机制的使用，那么如果一台主机宕机时可以保证数据不丢失。



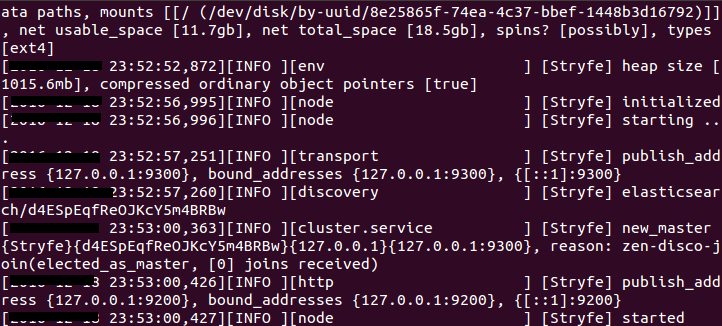
（a）ElasticSearch集群概览



(b) ES01节点进程启动



(c) ES02节点进程启动



（d） ES03 节点进程启动

图5-3 ElasticSearch集群启动情况

使用搭建的实验系统对IDS系统产生的安全告警日志进行了为期两天时间的采集。总共得到26431649条的安全日志。如图5-4是安全收集统计结果展示。结果按日期分为了两个Index。在经过系统采集处理后，日志被统一检索在ElasticSearch集群中。

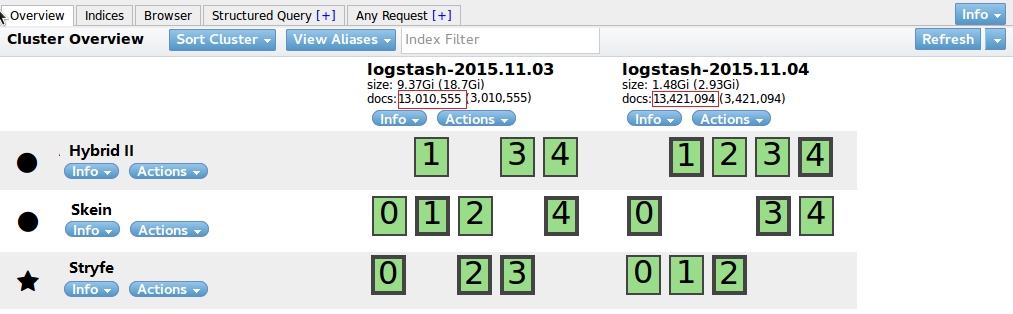


图5-4 安全日志收集结果

随机查询其中存储的一条日志信息，结果如图5-5所示，可见日志已经被正确的解析，并且为了减少不必要的属性对日志存储空间的占用，提高查询效率，日志的属性类型仅保留了源IP、目的IP、端口号、时间戳、以及攻击类型。经过后期统计，图5-6展示了安全日志攻击告警类型分布情况。

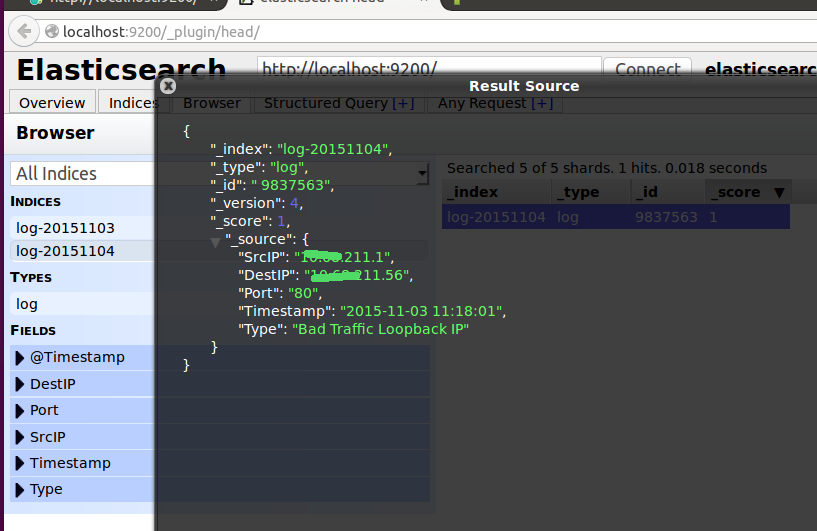


图5-5 安全日志存储格式

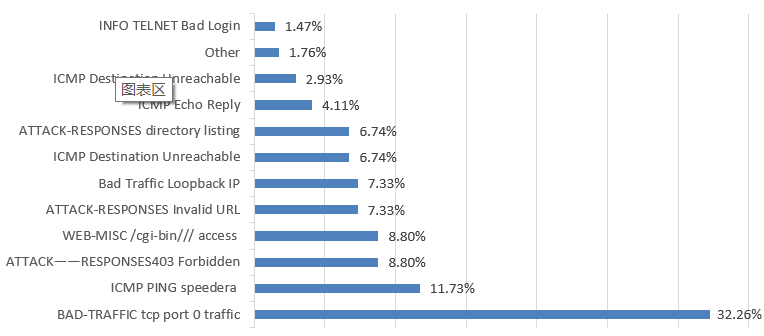


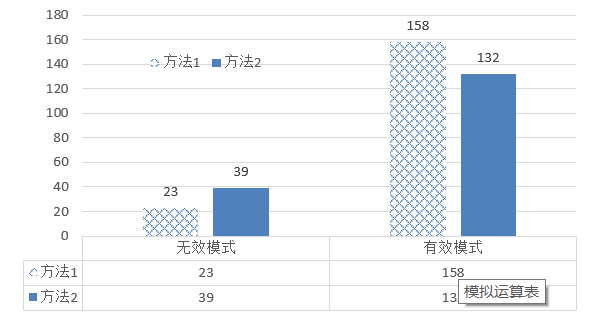
图5-6 安全日志告警类型分布

5.2.2 挖掘结果对比

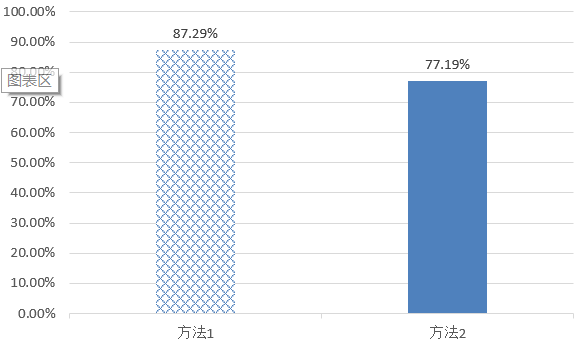
对于收集到的安全日志调用攻击模式挖掘模块进行攻击模式进行挖掘。同时为了验证本文方法（记为方法1）在实际情况下的性能，同样使用了文献[13]中的方法（记为方法2）进行对比实验，因此分别使用两种算法对数据集进行攻击模式挖掘。然后对挖掘结果进行人工分析校验，判断每一条结果是否是有意义的攻击模式。按照公式：

 （5-1）

用有意义的攻击模式数量和挖掘结果的总数量的比值来判断挖掘结果的正确率，正确率越高在利用攻击序列生成攻击模式的过程中人工花费越少[43]。结果统计如图5-7所示。可以看出方法攻击模式挖掘数量是181，其中有效的攻击模式158个，无效的攻击模式23个，准确率为87.29%；而方法2准确率为77.19%。本文算法准确率上有较大的提升，这样可以更加有效的分析数据挖掘结果。造成这种结果的原因是，通过模糊聚类分析算法根据日志的属性将日志进行聚类之后再对频繁攻击序列进行挖掘，能够一定程度上保证攻击场景的完整性，进而挖掘出的结果更加准确。



（a）结果数量对比



（b）准确率对比

图5-7 攻击模式挖掘结果对比

5.3 本章小结

本章首先搭建了系统测试环境，收集真实的安全日志，对系统运行状况进行了测试。然后以收集的安全日志数据为基础进行对比试验，从实验结果来看，论文所述挖掘算法能够较为准确的地从安全日志告警信息中发现代表真实入侵行为的攻击日志类型序列，能够更加准确的挖掘出有效的攻击模式。

第六章 总结与展望

6.1 本文工作总结

本文主要研究了基于安全日志的攻击模式挖掘技术。在研究了相关数据挖掘算法之后提出了基于模糊聚类分析和序列模式的攻击模式挖掘算法，并通过实验验证了该算法的可行性和有效性。另外本文还设计了一个基于ELK的攻击模式挖掘系统，并搭建实验环境对该系统进行测试。

本文完成的主要工作有：

（1）研究了相关的日志收集存储技术和数据挖掘相关理论与算法。

（2）提出一种基于改进模糊聚类分析和序列模式的攻击模式挖掘算法。该算法结合了模糊聚类能够描述安全日志间相似性和序列模式挖掘算法能够较好的描述攻击步骤间前后间逻辑关系的优势，可以更加准确的挖掘日志中的隐藏的攻击模式知识。方法首先将安全日志数据根据时间戳转化为全局攻击序列；然后利用改进的模糊聚类分析算法根据安全日志属性之间的相似度，将可能处在同一场景下的序列聚集在一起转化为攻击序列集；最后通过序列模式挖掘算法从攻击序列集中挖掘出攻击者的攻击模式。

（3）设计并搭建了基于ELK的攻击模式挖掘系统。该系统基于ELK技术可以完成从安全日志收集到攻击模式挖掘整个流程。

（4）搭建实验环境对算法进行分析评估，实验结果表明该算法不仅可以有效的挖掘出隐藏在安全日志中的攻击模式，还能在结果中减少无效的攻击模式，生成更有价值的攻击模式知识。

6.2 未来展望

本文虽然提出了一种基于模糊聚类分析和序列模式的攻击模式挖掘算法，但由于时间及各方面条件的限制,本文提出的算法还未在大规模实践中得到验证，这也是下一步着重需要解决的问题。同时在基于安全日志的攻击模式挖掘领域中还有一些问题需要进一步的研究解决。

（1）安全日志不论是来源还是属性维度都趋向于多元化，如何集中统一各种设备的日志信息并充分利用日志中的每个属性进行挖掘需要进一步的研究。

（2）本文中所提出攻击模式挖掘算法,仅仅考虑到历史安全日志,每一次挖掘都需要对全局或者一段时间之内的安全日志进行分析，没有考虑到日志的增量。如何利用增量日志，来进行攻击模式挖掘这也成为序列模式挖掘算法研究的一个发展方向。

（3）伴随着数据集的急剧增长，算法的运行时间也在不断增加，对于攻击模式挖掘算法如何应用在分布式计算框架中以解决算法计算效率问题，也需要在未来的研究中不断探索。

参考文献

1. 国家计算机网络应急处理协调中心. 2015年中国互联网网络安全报告[R]. 北京: 人民邮电出版社, 2015.
2. Kokila RT, S. Thamarai Selvi, K. Govindarajan. DDoS detection and analysis in SDN-based environment using support vector machine classifier[A]. // 2014 Sixth International Conference on Advanced Computing (ICoAC)[C], 2014: 205-210.
3. Templeton Steven J, Levitt K. A requires/provides model for computer attacks[A]. // In Proceedings of the 2000 workshop on New security paradigms[C], 2000: 31-38.
4. Ning P, Cui Y, Reeves D. S. Constructing attack scenarios through correlation of intrusion alerts[A]. // In Proceedings of the 9th ACM conference on Computer and communications security[C], 2002: 245-254.
5. Cuppens F. Managing Alerts in a Multi-Intrusion Detection Environment[A]. // In Proceedings of the 17th Annual Computer Security Applications Conference[C], 2001: 22-31.
6. Cuppens F, Miege A. Alert correlation in a cooperative intrusion detection framework[A]. // In Proceedings of the 2002 IEEE symposium on security and privacy[C], 2002: 202-215.
7. Cuppens F, Autrel F, Miege A, et al. Correlation in an intrusion detection process[A]. // In Proceedings of the SECI02 Workshop[C], 2002: 153-171.
8. Qin X, Lee W. Statistical causality analysis of INFOSEC alert data[A]. // In Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection[C], 2003: 591-627.
9. Qin X, Lee W. Discovering novel attack strategies from INFOSEC alerts[A]. // In Proceedings of the 9th European Symposium on Research in Computer Security[C], 2004: 439-456.
10. Qin X, Lee W. Causal discovery-based alert correlation[A]. // In Proceedings of the 21st Annual Computer Security Application Conference[C], 2005: 33-40.
11. Zhu B, Ghorbani A A. Alert Correlation for Extracting Attack Strategies[J]. International Journal of Network Security, 2006, 3(3)(3).
12. Kavousi F, Akbari B. A Bayesian network-based approach for learning attack strategies from intrusion alerts[J]. Security & Communication Networks, 2014, 7(7): 833–853.
13. A. f. Zhang, Z. t. Li, D. Li and L. Wang. Discovering Novel Multistage Attack Patterns in Alert Streams[A] // 2007 International Conference on Networking, Architecture, and Storage (NAS 2007)[C], 2007: 115-121.
14. Hellerstein J L, Ma S. Mining Event Data for Actionable Patterns.[A] // International Computer Measurement Group Conference[C], 2000: 307-318.
15. J. J. Treinen, R.Thurimella. A Framework for the Application of Association Rule Mining in Large Intrusion Detection Infrastructures[J]. Recent Advances in Intrusion Detection, 2006: 23-38.
16. 杨尚大. 日志数据采集和实时审计关键技术研宄与实现[D] 浙江: 浙江工商大学, 2011
17. 郑礼良. 基于入侵检测和日志审计的网络安全方案[D]. 安徽: 合肥工业大学, 2011
18. 段娟,辛阳,马宇威. 基于Web应用的安全日志审计系统研究与设计[J]. 信息网络安全. 2014(10)
19. 申彦. 大规模数据集高效数据挖掘算法研究[D]. 江苏: 江苏大学, 2013.
20. 何清,庄福振. 基于云计算的大数据挖掘平台[J]. 中兴通讯技术, 2013, 04:32-38.
21. H. Li, S. Chen, J. Li, S. Wang and Y. Fu. An improved multi-support Apriori algorithm under the fuzzy item association condition[A], // 2011 International Conference on Electronics, Communications and Control (ICECC)[C], 2011:3539-3542.
22. X. Yu and T. Korkmaz. Super-sequence frequent pattern mining on sequential dataset[A], // 2013 IEEE International Conference on Big Data, Silicon Valley[C], 2013:52-59.
23. T. Xinguang, D. Miyi, S. Chunlai and L. Xin. Detecting network intrusions by data mining and variable-length sequence pattern matching[J]. Journal of Systems Engineering and Electronics, 2009, 20(2):405-411.
24. 刘彤. 基于Hadoop的数据分析系统的设计与实现[D]. 北京:北京邮电大学, 2012.
25. D. Bernstein. The Emerging Hadoop, Analytics, Stream Stack for Big Data[J]. IEEE Cloud Computing, 2014, 1(4): 84-86.
26. Xiuqin Lin，Peng Wang，Bin Wu.Log analysis in cloud computing environment with Hadoop and Spark[A]. // 5th IEEE International Conference on Broadband Network & Multimedia Technology[C], Piscataway: IEEE Press, 2013: 273-276.
27. 朱珠. 基于Hadoop的海量数据处理模型研究与实现 [D]. 北京:北京邮电大学, 2008.
28. U. Thacker, M. Pandey and S. S. Rautaray. Performance of elasticsearch in cloud environment with nGram and non-nGram indexing[A], // 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)[C], Chennai: IEEE, 2016: 3624-3628.
29. J. Bai. Feasibility analysis of big log data real time search based on Hbase and ElasticSearch[A]. 2013 Ninth International Conference on Natural Computation (ICNC)[C], Shenyang, 2013: 1166-1170.
30. A. Singh and H. G. Vélez. Hierarchical Multi-log Cloud-Based Search Engine[A], // 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems[C], Birmingham, 2014: 211-219.
31. Y. Zhang and J. l. Li. Research and Improvement of Search Engine Based on Lucene[A]. 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics[C], Hangzhou 2009: 270-273
32. A. Lahmadi, F. Beck, E. Finickel and O. Festor. A platform for the analysis and visualization of network flow data of android environments[A], // 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)[C], Ottawa,: IEEE, 2015: 1129-1130.
33. 刘俊龙,刘光明,张黛,喻杰. 基于Redis的海量互联网小文件实时存储与索引策略研究[J]. 计算机研究与发展, 2015, S2: 148-154.
34. 李诗云. 基于内存数据库Redis的众包系统性能优化[D]. 浙江: 浙江大学, 2016.
35. 陈韶男. 基于NoSQL的综合信息汇聚平台的设计与实现[D]. 北京: 北京邮电大学, 2015.
36. 魏为民,袁仲雄. 网络攻击与防御技术的研究与实践[J]. 信息网络安全, 2012, 12: 53-56.
37. SergiosTheodoridis, KonstantinosKoutroumbas, 西奥多里迪斯,等. 模式识别 [M]. 第二版。 北京: 电子工业出版社, 2004.
38. 朱林,王士同,邓赵红. 改进模糊划分的FCM聚类算法的一般化研究[J]. 计算机研究与发展, 2009, 05: 814-822.
39. 于洋. 模糊聚类分析中模糊c均值聚类计算方法研究[D]. 沈阳: 沈阳工业大学, 2009.
40. 石陆魁,何丕廉,柳冰等. 一种快速的模糊聚类算法[A].// 2005年中国模糊逻辑与计算智能联合学术会议论文集[A]. 2005:347-350.
41. MIT Lincoln Laboratory DDoS 1.0 Intrusion Detection Dataset [DB/OL].http://www.ll.mit.edu/IST/ideval/data/2000/LLS\_DDOS\_1.0.html.
42. 2000 DDo S 2.0.2 Intrusion Detection Dataset Host [EB/OL]. http://www.ll.mit.edu/IST/ideval/docs/2000/2000\_LLS\_DDOS\_2.0.2\_hosts.html.
43. 朱林. 基于特征加权与特征选择的数据挖掘算法研究[D].上海交通大学,2013.

致 谢

论文即将完成，预示着我的两年半的研究生生涯就要结束，十九年的求学生涯也即将终结。回首这两年半的研究生生活，发现有太多的人需要感谢，借此机会，向所有关心帮助我的人表示感谢。

首先，我要感谢我的导师刘建毅老师和张茹老师。论文从开题到最终完成，都离不开两位老师细心的指导和不懈的支持。张老师和蔼的教诲给我指明了努力的方向。刘老师严谨、富有智慧的言谈帮我度过了科研项目中的种种困境。谢谢他们，祝两位老师身体健康、工作顺利。

感谢实验室的李芙蓉、王俊、许裔、薛颖、曾恒同学，有幸和他们一起学习奋斗是我读研最重要收获。和他们一起玩桌游的时光总是充满了欢乐，谢谢他们，让我的研究生生活变得如此丰富多彩。感谢魏金侠师姐，师姐严谨治学的态度、乐于助人的品格一直是我学习的榜样。感谢实验室的师弟师妹们，和他们在同一个实验室中一起学习、一起奋斗的日子将是我人生中宝贵的记忆。

感谢陈四通、胡逸云、王龙华、徐昊、许裔同学，感谢学十637的舍友们，在科研之外的闲暇时间，和他们一起聊人生，谈理想将是我最难忘的回忆。

还要感谢汤宝铮、许丞他们不仅是我的双重校友，更是我生活中的挚友，有他们陪伴，永远不会感到孤单。

最后感谢我的家人。感谢父母，二十五年来，是他们的默默支持让我不断前进，是他们的教导让我成长，是他们让我知道如何成为一个有担当的人。感谢邢丹，是她的陪伴让我度过了一个又一个困难，和她一起努力奋斗的日子永远阳光灿烂。

告别学生时代，走向职场，我将乘风破浪、扬帆远航！

攻读学位期间发表的学术论文

1. Y. Li, Y. Xue, Y. Yao, X. Zhao, J. Liu and R. Zhang, "An attack pattern mining algorithm based on fuzzy logic and sequence pattern[A], // 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)[C], Beijing, China, 2016, pp. 234-238. doi: 10.1109/CCIS.2016.7790260.
2. Y. Xue, Y. Li, Y. Yao, X. Zhao, J. Liu and R. Zhang, Phishing sites detection based on Url Correlation[A], //2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)[C], Beijing, China, 2016, pp. 244-248. doi: 10.1109/CCIS.2016.7790262.