doi:10.11959/j.issn.1000-436x.2017143

基于多属性决策及污点跟踪的大数据平台敏感信息泄露感知方法

沙乐天[1,2]，肖甫[1,2]，陈伟[1,2]，孙晶[3]，王汝传[1,2]

（1. 南京邮电大学计算机学院，江苏 南京 210023；
2. 江苏省无线传感网高技术研究重点实验室，江苏 南京 210023；3. 南京电讯技术研究所，江苏 南京 210007）

摘 要：基于多属性决策及污点跟踪提出一种面向大数据平台中敏感信息泄露的感知方法，该方法通过分析已知大数据平台敏感信息泄露的相关已知漏洞，抽取并推演目标敏感信息集合，并结合敏感信息操作语义建立目标集多属性模型，进而设计基于灰色关联分析及理想优基点法的敏感度计算方法，并基于污点跟踪实现了原型系统，最终实现了基于所提方案的跨平台敏感信息泄露漏洞的挖掘与验证。实验表明，所提方法可有效实现敏感信息泄露场景的已知漏洞验证及未知漏洞挖掘，从而为敏感信息动态数据流的安全防护提供支持。

关键词：多属性决策；污点跟踪；大数据平台；敏感信息

中图分类号：TP393 **文献标识码**：A

Sensitive information leakage awareness method for big data platform based on multi-attributes decision-making and taint tracking

SHA Le-tian[1,2], XIAO Fu[1,2], CHEN Wei[1,2], SUN Jing[3], WANG Ru-chuan[1,2]

(1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, China;
3. Nanjing Telecommunication Technology Institute, Nanjing 210007, China)

Abstract: Based on multiple-attribute-decision-making and taint tracking,a sensitive-information leakage awareness method was proposed, some relative known vulnerabilities in big data platform was analyzed, target database was extracted and extended, multiple attribute model was built combined with operation semantic, a grey-correlation-analysis and technique for order preference by similarity to an ideal solution based sensitivity measurement was designed in combination of regular operation semantic for sensitive information. A prototype was built based on taint tracking, sensitive-information leakage vulnerabilities could be verified and discovered across big data platforms in this method. The experiment shows that verification for known bugs and discovery for unknown vulnerabilities can be accomplished based on leakage scenarios, which can be regarded as a support for protection in dynamic sensitive information data flow.

Key words: multi-attributes decision making, taint tracking, big data platform, sensitive information

1 引言

云计算、物联网、移动互联网等新兴信息技术和应用模式的快速发展，促使全球数据量急剧上升，大数据迅速发展成为工业界、学术界甚至世界各国政府关注的热点[1~4]。目前，制约云计算与物联网广泛应用的关键点就在于数据安全与隐私保护，而移动互联网背景下的恶意攻击与敏感信息窃取更是层出不穷。在此背景下，大数据安全分析技术得到业界广泛认可，其高速的流式处理方法可有

收稿日期：2016-12-23；**修回日期**：2017-03-14
通信作者：肖甫，xiaof@njupt.edu.cn
基金项目：国家自然科学基金资助项目（No.61373137）；江苏省高校自然科学研究计划重大基金资助项目（No.14KJA520002）；江苏省自然科学基金资助项目（No.BK20161516）

Foundation Items: The National Natural Science Foundation of China (No. 61373137), Major Program for Natural Science Foundation of Jiangsu Higher Education Institutions (No.14KJA520002), The Nature Science Foundation of Jiangsu Province (No.BK20161516)

效适用于 DDoS 和 CC 等传统恶意攻击检测，同时，对高级可持续性威胁（APT, advanced persistent threat）有较好的在线检测效果[1,2]；对异构海量数据的离线处理可应用于云计算或物联网模式下的隐私识别与防护[3,4]。

由此可见，大多数大数据分析平台的安全相关工作主要是利用大数据平台完成安全分析或数据处理，却忽略了大数据分析平台本身的安全防护。基于大数据平台的开放性和多元化，通常会为用户提供二次开发的调用接口或脚本运行的加载方式，同时，也需要保存或校验用户认证过程中生成的各种中间数据。此类数据的泄露将造成大数据平台被攻击者远程控制或用户隐私等关键信息曝光，从而造成不可挽回的损失。近年来，CVE 记录的若干列表项均属于此类漏洞，如 CVE-2012-3376、CVE-2015-1776 和 CVE-2015-3188 等。

综上，本文提出一种针对大数据平台的敏感信息泄露感知方法，根据预定义的函数、脚本接口及加密认证的中间数据推演生成目标测试数据集，抽取常规的操作规则为多属性决策提供参考准则，采用灰色关联分析与理想优基点结合的方法来计算敏感度及阈值，并通过污点跟踪方法实现原型系统，最终实现了本文方案下的跨平台相关类型漏洞的挖掘与验证。本文主要贡献如下：1) 从逆向分析者的角度理解几种主流的大数据平台中敏感信息的动态数据流向，定位并推演敏感信息集合；2) 从多属性聚合的角度完善敏感信息的定义，结合敏感信息处理语义的抽取，定义全局的大数据平台敏感信息，便于实现敏感信息出现点的关联性度量；3) 根据基本属性及语义对敏感信息数据流向进行聚类，采用灰色分析与理想优基点相结合的方法取得敏感度，实现敏感信息的可信性度量，为后续的敏感信息保护工作及相关类型漏洞挖掘提供理论依据。

2　研究现状

当前，主流的大数据安全技术主要从控制流和数据流上来保障网络环境的安全性与敏感信息的保密性，如图 1 所示，主要涉及以下 2 个方面。

1) 大数据安全分析技术。现有研究工作中，研究者主要通过改进、优化机器学习、数据挖掘等算法针对各类恶意攻击实例展开安全分析，主要目的在于面向大数据环境发现恶意攻击行为。尤其针对当前复杂网络环境下的 APT 攻击行为，在应用层从流量、主机、服务器和移动端等角度展开事件挖掘，在 APT 攻击的侦查准备、初次入侵和保持访问等阶段通过大数据分析挖掘出细粒度的事件特征及事件关联[1]，在内核层基于云环境下的虚拟化技术部署内核钩子，通过 VMM 的虚拟化环境切换完成事件响应[2]，为安全防御提供支持。同时，文献[3]以大数据的分析方法为切入点，详细阐述了大数据来

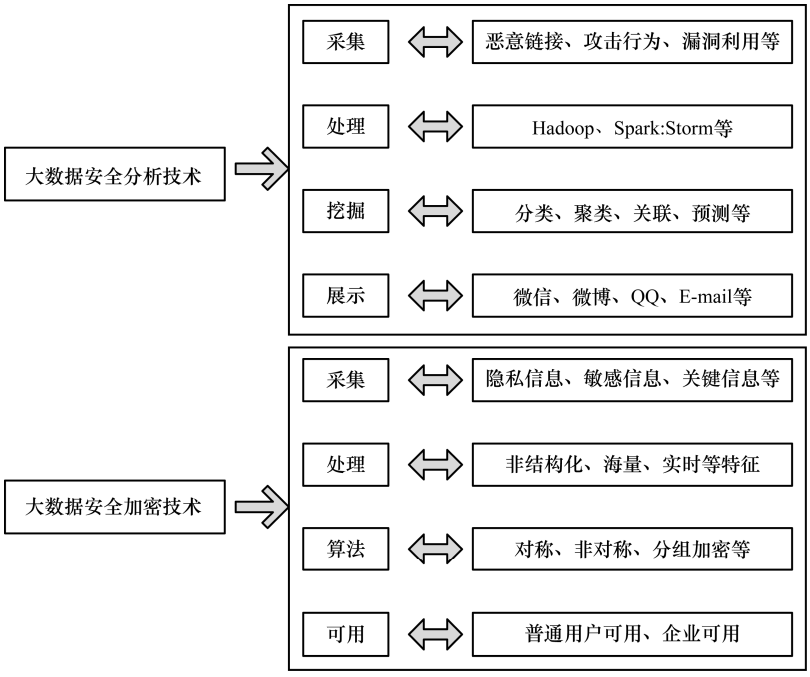


图 1　当前主流大数据安全技术

源及其特征、大数据分析目标、大数据技术架构，从大数据环境中的用户隐私、大数据的可信性和大数据环境中的访问控制模型等角度展示和挖掘相关安全问题，为大数据环境中的用户隐私分析提供参考。文献[5]则是面向大数据环境下的用户个人隐私展开数据特征分析，分析大数据背景下在数据层、应用层、数据展示层中个人隐私数据的累积性和关联性，展示大数据环境下隐私泄露的威胁程度。

2) 大数据安全加密技术。为保障企业及政府机构的大数据本身的安全，当前研究者主要提供面向大数据的加密、存储和检索等技术，在保障大数据可用性的前提下提升大数据的保密性[6~11]。文献[6]是基于大数据环境中隐私分析的保护技术综述，从密码学角度阐述了隐私保护的研究进展及主要方向，针对大数据的存储、搜索和计算 3 个方面给出研究背景介绍，包括大数据完整性审计协议、大数据密文检索以及大数据安全计算等，并给出基于数据恢复功能的完整性审计协议、支持动态结构的安全搜索以及高效全同态加密的研究方向。文献[7]则从大数据存储的角度设计了一种安全增强的存储接口和协议，可以为用户提供面向大数据存储的细粒度访问控制，并通过 HDFS 的对比来验证系统效率。文献[8]从安全多方计算的角度描述了完全同态加密方法的实现过程，具体应用到大数据环境中，用户可以请求共有云调用求值算法对密文进行操作，而后将计算结果返回给用户，最终用户利用私钥进行解密得到期望的结果，从而实现用户隐私数据的防泄露。文献[9]则从云环境中大数据安全去重的角度分析了近些年相关的研究，尤其指出基于内容加密安全去重，通过对密文的证明实现文件级和本地数据的重复性检测[10]，可实现客户端机密数据安全去重并抵抗对数据块的暴力搜索攻击。文献[11]证明通过半同态和全同态加密算法可对数据密文进行部分操作，但计算开销较大、系统效率较低。因此，如何对大数据执行密文的安全匹配、重复校验等操作，并取得安全去重、隐私保护和效率之间的平衡，是当前的一个研究难点。

以上 2 种大数据安全技术从安全分析及安全加密的角度完成了大数据平台在安全研究领域的应用及用户隐私保护，但未考虑大数据安全分析或安全加密平台本身的安全性，尤其是基于开源工具的二次开发平台安全性。根据互联网周刊发布的《2015 年 Q1 中国大数据分析工具 TOP30 排行榜》[12]，目前，主流的大数据分析工具主要包括 IBM 公司的 InfoSphere[13]、Google 公司的 BigQuery[14]和 Amazon 公司的 Kinesis[15]等。从工具基本架构上看，离线分析大都基于 Hadoop[16]平台的二次开发实现，在线处理则是在 Storm[17]或 Spark[18]平台的基础上优化生成。目前，尚没有针对大数据平台的安全防护系统，大数据应用企业对目标数据进行分析处理的过程中只是采用传统的 IDS、UTM 等部署入侵检测或安全防御。若攻击者通过开源底层系统研究获得 0day 未知漏洞，即可非法读写分析平台中的各种敏感信息，如敏感函数调用、用户认证中间数据等，造成控制流劫持或用户身份仿冒。目前，典型的大数据平台中的信息泄露类漏洞包括 CVE-2012-3376（Hadoop 中使用 Kerberos 认证过程中出现错误导致远程用户可读写任意 block 敏感信息）、WooYun-2013-22434（大量 Hadoop 应用对外访问所导致的敏感信息泄露）、CVE-2015-1776（Hadoop 中特殊功能开启时本地用户可读取加密 MapReduce 的凭证文件）；同时，还有导致外部代码任意执行类漏洞：CVE-2015-3188（storm 中 UI 守护程序存在远程代码执行漏洞，可使远程用户以当前用户权限运行任意代码）。后者同样可导致信息窃取类的恶意代码执行。

本文方案基于大数据分析平台本身的架构及特征，定义平台在运行过程中所生成的各种敏感信息，结合污点跟踪过程中的内存页标记实时获取敏感信息泄露场景及恶意篡改行为。对比目前已有的面向大数据环境的访问控制模型[12,13]或用户隐私保护方法[3,5,6]，本文研究工作涉及大数据分析平台中的敏感信息定义、抽取、跟踪及动态安全度量，在此基础上实现感知敏感信息泄露的原型系统。

3 基于多属性决策及污点跟踪的泄露感知

3.1 属性定义

面向大数据的分析平台在运行过程中涉及 2 类主要数据：1) 被分析用户或系统的目标数据；2) 分析平台使用者的管理数据。鉴于数据本身的属主特征，敏感信息泄露问题通常针对第 2) 类数据衍生。总结已有漏洞的敏感信息泄露场景及大数据分析平台本身的特质，本文方案将大数据分析过程中所产生的敏感信息汇聚如图 2 所示。信息源主要来自于系统记录、网络流量、文件系统等；数据采集主

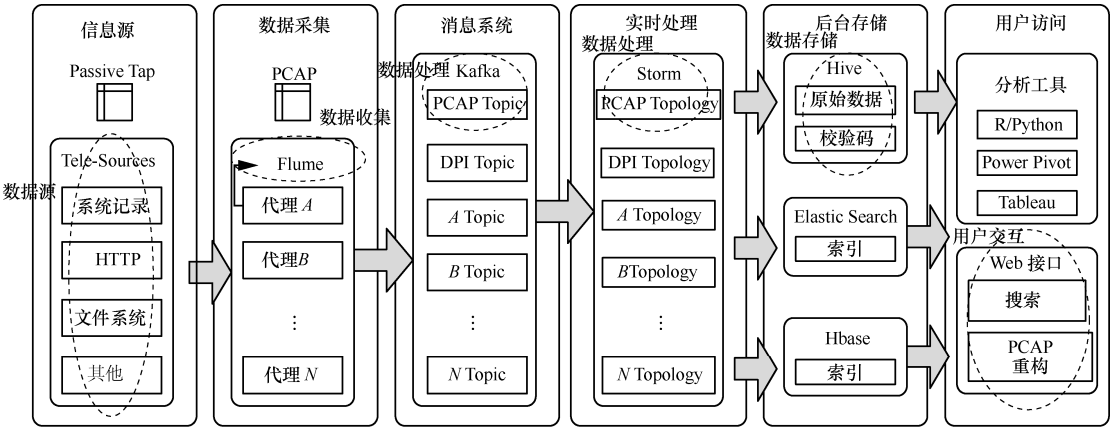


图 2　大数据分析平台中各层敏感信息集合

要指 Flume 系统中多代理节点的收集信息；消息系统以 Kafka 为典型代表，包括 PCAP、DPI 等多个 Topic 中的消息传递；数据的实时处理及后台存储则以 Storm 和 Hive 为典型对象进行分析，经 Topology 带 Hive 节点完成数据流处理并存储。最终用户可通过一些简单的脚本语言，如 R 或 Python 访问相关接口并进行数据搜索等处理。

大数据平台中的关键组件具体包括数据源层的数据定位与采集、数据处理中的聚类方法与实时分析、数据存储中数据读写以及用户交互过程中的数据访问等。而结合已出现的大数据分析平台相关漏洞，可将敏感信息分成以下几类。1) 开放 API：基于开放端口的访问通常需要结合开放 API 的调用才能实现有目的性的特权操作；2) 开放脚本：通过创建存储数据或表项可自定义脚本命令实现任意代码执行；3) 数据加密或用户认证的中间数据：不恰当的中间数据存储或访问规则可导致此类数据泄露，为攻击者仿冒或后继攻击提供帮助。

针对以上几类敏感信息，本文方案重点关注其在大数据平台内部的处理过程，其中涉及敏感信息的创建、校验、复制、加密和回收等操作。从各类敏感信息在大数据平台中的动态数据流中可以抽取敏感信息处理的固有属性及可变属性，结合各属性的特征以及权重分配方法，可在敏感信息动态流向中的不同敏感信息出现点处进行动态敏感度度量。采用个体敏感数据（ISD，individual sensitive data）描述敏感信息出现点，全局处理流程可表示为个体敏感数据的集合，即 $GSD=\{ISD_{(1)}, ISD_{(2)},\cdots, ISD_{(n)}\}$，随后给出数据属性相关的定义。

定义 1　$ISD.IA=\{ISD.T, ISD.\ Addr, ISD.F, ISD.P\}$。下面，分别给出其属性元素的描述。

$ISD.T=\{Func, Scr, IntD\}$，用于描述敏感信息的 3 种类型：函数（开放的 API，简记为 *Func*）、脚本（可调用的脚本运行接口，简记为 *Scr*）、中间数据（加密或认证过程中的可见中间数据，简记为 *IntD*）。

$ISD.Addr=\{U, V, VBU, W\}$，描述定位该敏感信息的内存地址。对函数类敏感信息，基于 Java 开发框架的大数据平台在函数调用过程中通常不需要计算函数地址，同时，也不需要关注函数指针的内存位置，但基于函数的操作必须保证函数地址是可见的（visible，简化为 *V*），即目标函数可调用，否则函数或方法调用不可行（unvisible，简化为 *U*）。而对中间数据类敏感信息，内存地址是否可写是判决泄露程度的关键因素之一，可读但不可写（visible but unwritable，简化为 *VBU*）是中度泄露。在影响最大的情况下，对函数或中间数据，地址处可写（writable，简化为 *W*），则可对数据篡改。对脚本类敏感信息，不存在内存地址的概念。

$ISD.F=\{P, C, OF\}$，用于描述敏感信息在局部出现点的 3 种数据表现形式：明文（*P*）、密文（*C*）和其他（*OF*）。*OF* 是指对敏感信息特殊处理后所得到的数据形式，如将其全部置零等。而密文则根据加密的过程可细化为不同安全级别的密文。对函数或脚本类敏感信息，密文主要表现为函数或脚本调用点的混淆程度，而对中间数据而言密文具体指加密算法处理后的数据。

$ISD.P=\{OU, IU, IA\}$，用于描述敏感信息本身所代表的特权等级。对大数据分析平台来说，特权等级通常可分为外部用户级（*OU*）、内部用户级（*IU*）、内部管理员级（*IA*），权限级依次递增。

定义 2 *ISD.VA*={*ISD.FI*, *ISD.SR*, *ISD.IC*}。下面，分别给出其属性元素的描述。

ISD.FI={*N*, *CoFI*, *CrFI*}，用于标记函数类敏感信息在传播过程中是否存在动态调用效果。大体可分为3类：1) 未检测到敏感函数被动态调用（None，简化为 *N*）；2) 检测到敏感函数被动态调用，但不存在恶意的函数调用效果（CommonFuncInvoking，简化为 *CoFI*）；3) 检测到敏感函数被调用，且存在恶意的函数调用效果（CriticalFuncInvoking，简化为 *CrFI*）。函数调用效果的恶意判决方法将在第 4 节中具体描述。

ISD.SR={*N*, *CoSR*, *CrSR*}，用于标记脚本类敏感信息在传播过程中是否存在动态调用效果，具体赋值原理同 *ISD.FI*，此处不再重复。

ISD.IC={*NR*, *RBNW*, *W*}，用于评估用户身份相关的敏感数据泄露时所导致身份仿冒的恶意效果，大体可分成：1)所泄露的中间数据不可读（*NR*）；2) 所泄露的中间数据可读但不可写（*RBNW*）；3) 所泄露的中间数据可写（*W*）。具体的判决标准将在第 4 节中描述。

定义 3 *ISD.SL* 描述敏感信息的动态敏感度。该值基于多属性决策方法从以上定义中汇聚而成，可以表示敏感信息在大数据平台中的动态安全级别，用于衡量敏感信息泄露的可能性及泄露后造成的风险。第 5 节将介绍该值的计算。

针对敏感信息在应用软件中的每一个出现点，都可以求得该点个体敏感数据的所有属性值，根据这些属性值，可以给出一个全局敏感信息数据处理(GSD, global sensitive data)的属性描述。

定义 4 *GSD.Attribute*={*GSD.T*, *GSD. BP*, *GSD.DP*, *GSD.ST*}。*GSD.T*{*Func*, *Scr*, *IntD*}，该属性与 *ISD.T* 相同；*GSD.BP* 描述目标敏感信息首次被定位时的位置，对于函数或脚本类，主要记录其与端口的绑定条件，对于中间数据，则记录其内存地址；*GSD.DP* 描述敏感信息回收时的位置，对于函数类或脚本类，该定义描述在不同版本中补丁升级后的不可调用效果，对于中间数据，该定义则主要描述平台内部如何销毁目标数据；*GSD.ST* 描述敏感信息预泄露时的动态敏感度，通过该值的比较可判决动态敏感度是否超过安全阈值。第 5 节将介绍该值的计算。

3.2 多属性决策的语义规则

面向大数据分析平台部署污点分析和内存切片，本文方案可以获得敏感信息各出现点的固有属性和可变属性。而面向敏感信息的各类操作属于高层语义，如函数或脚本调用、中间数据的读写等，固有属性和可变属性则属于低层语义。因此，根据经验和实践研究，总结和设计从低层语义映射到高层语义的规则，依次为基础构建决策过程中的多属性判决标准。该规则基本可覆盖大数据分析平台本身希望提供给普通用户或管理员用户的各类正常数据操作方法，而实际的敏感信息数据流向中若出现常规语义之外的操作过程，则代表了异常或违规的敏感数据操作。该映射规则的正确性经若干种主流大数据分析平台的动态污点跟踪及源码反编译方法所验证，从而确保推求所得的语义规则的合理性。

规则 1 对于函数类敏感信息，若函数调用可见，函数名称开放，可为外部用户或内部用户提供普通函数调用接口。

$$\exists ISD_{(i)}.T = Func$$
$$\&\& ISD_{(i)}.Addr = V$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = OU \mid IU$$
$$\Rightarrow ISD_{(i)}.FI = CoFI$$

规则 2 对于函数类敏感信息，若函数调用可见，函数名称开放，则可为内部管理员提供关键函数调用接口。

$$\exists ISD_{(i)}.T = Func$$
$$\&\& ISD_{(i)}.Addr = V$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = IA$$
$$\Rightarrow ISD_{(i)}.FI = CoFI \mid CrFI$$

规则 3 对于脚本类敏感信息，若脚本调用接口可见，则可为外部用户或内部用户提供普通函数调用接口。

$$\exists ISD_{(i)}.T = Scr$$
$$\&\& ISD_{(i)}.Addr = V$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = OU \mid IU$$
$$\Rightarrow ISD_{(i)}.SR = CoFI$$

规则 4 对脚本类敏感信息，若脚本调用接口可见，可为内部管理员用户提供关键函数调用接口。

$$\exists ISD_{(i)}.T = Scr$$
$$\&\& ISD_{(i)}.Addr = V$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = IA$$
$$\Rightarrow ISD_{(i)}.SR = CoFI \mid CrFI$$

规则 5　对于中间数据类敏感信息，若地址可读但不可写，数据格式为明文或其他格式，则可为内部管理员用户提供可读操作接口。

$$\exists ISD_{(i)}.T = IntD$$
$$\&\& ISD_{(i)}.Addr = VBU$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = IA$$
$$\Rightarrow ISD_{(i)}.IC = RBNW$$

规则 6　对于中间数据类敏感信息，若地址可读且可写，数据格式为明文或其他格式，则可为内部管理员用户提供可写操作接口。

$$\exists ISD_{(i)}.T = IntD$$
$$\&\& ISD_{(i)}.Addr = VBU$$
$$\&\& ISD_{(i)}.F = P \mid OF$$
$$\&\& ISD_{(i)}.P = IA$$
$$\Rightarrow ISD_{(i)}.IC = W$$

3.3　泄露感知

针对每一类敏感信息的操作语义，需根据其所对应的不同属性赋予不同权重，生成归一化的评价聚合值，本文方案采用多属性决策理论完成该过程。多属性决策是在具有相互冲突、不可判决的多个属性情况下取得最优解决方案的基本方法，具体来看，根据对象属性特征的区别分成定量及定性信息的多属性决策评估[14]。本文方案中各属性均有量化标准，因而，采用定量信息的多属性决策模型。同时，本文方案需要基于定量多属性决策生成综合指标用以判决实时敏感信息动态流向中的敏感度，并与敏感度阈值进行比较从而衡量其泄露程度与风险。因此，选择了理想优基点法[15]。其原始算法是以理想解与反理想解为参照基准，其中，每一属性中极大集聚合生成理想优基点，而极小集聚合生成反理想优基点，并采用欧几里得距离来计算目标决策方法与两者之间的差值，以此来评价决策的优劣。但针对本文方案中有限的属性集合和决策集合，且各属性值之间不确定的关联关系，使用该传统算法难以保证决策结果的正确性和准确性。而灰色关联分析是挖掘数据内部不确定关联的有效方法。灰色关联分析基于灰色关联度，以数据序列的几何接近度分析并确定因子之间的影响程度[16]。灰色关联分析的基本思想是对数据序列几何关系和曲线几何形状的相似程度进行比较分析，以曲线间相似程度大小作为关联程度的衡量尺度。曲线越接近，相应序列之间的关联度越大；反之则越小。灰色关联分析是对关联关系不清晰或根本缺乏事物关联原型的灰关系序列化、模式化，进而建立灰关联分析模型，使灰关系量化、序化、显化，能为复杂系统的建模提供重要的技术分析手段。因此，本文方案引入一种基于灰色关联度和欧几里得距离的多属性决策方法，以此对大数据分析及处理平台中出现的几类敏感信息进行敏感度度量。

3.3.1　面向属性值的去量纲化处理

为实现面向各属性的归一化的定量赋值，首先要针对不同类型敏感信息的固有及可变属性设定统一标度，以此去除不同属性所具备不同量纲对最近聚类分析结果的影响。定义各属性对敏感信息敏感度的影响因子为 *Factor*(*ISD.A*)，具体方法为选择 10 点标度作为量化范围，对影响敏感信息安全度最大的属性值赋 9.0，标记为 $F^{\max}(ISD_{(i)}.A)$，相反，对影响最小的属性值赋 1.0，标记为 $F^{\min}(ISD_{(i)}.A)$，以此为基础进行均匀量化。进而采用极差变换法完成去量纲化处理。均匀量化后的影响因子标记为 $Factor^{s}(ISD_{(i)}.A)$，具体过程如式(1)所示，结果如表 1 所示。

$$Factor^{s}(ISD_{(i)}.A) = \frac{F(ISD_{(i)}.A) - F^{\min}(ISD_{(i)}.A)}{F^{\max}(ISD_{(i)}.A) - F^{\min}(ISD_{(i)}.A)} \tag{1}$$

3.3.2　建立决策矩阵

建立敏感信息决策属性集。针对敏感数据处理的各个局部处理流程，选择 $ISD_{(i)}$为决策对象，以元素可变属性为决策对象属性构建决策目标矩阵

$$\exists x^{i} = ISD_{(i)}, x_{j}^{i} = ISD_{(i)}.A,$$
$$\boldsymbol{D}^{i} = (x_1^{i}, \cdots, x_6^{i})^{\mathrm{T}} = (ISD_{(i)}.Addr, ISD_{(i)}.F, ISD_{(i)}.P, ISD_{(i)}.FI, ISD_{(i)}.SR, ISD_{(i)}.IC) \tag{2}$$

表 1 量化后的标准化属性影响因子

$ISD.A$	量化后标准化影响因子
$ISD_{(i)}.Addr$	U: $Factor^S(ISD_{(i)}.A)$=0.1
	V: $Factor^S(ISD_{(i)}.A)$=0.325
	VBU: $Factor^S(ISD_{(i)}.A)$=0.5
	W: $Factor^S(ISD_{(i)}.A)$=0.9
$ISD_{(i)}.F$	P: $Factor^S(ISD_{(i)}.A)$=0.9
	C: $Factor^S(ISD_{(i)}.A)$=0.1
	OF: $Factor^S(ISD_{(i)}.A)$=0.5
$ISD_{(i)}.P$	OU: $Factor^S(ISD_{(i)}.A)$=0.9
	IU: $Factor^S(ISD_{(i)}.A)$=0.5
	IA: $Factor^S(ISD_{(i)}.A)$=0.1
$ISD_{(i)}.FI$	N: $Factor^S(ISD_{(i)}.A)$=0.1
	$CoFI$: $Factor^S(ISD_{(i)}.A)$=0.5
	$CrFI$: $Factor^S(ISD_{(i)}.A)$=0.9
$ISD_{(i)}.SR$	N: $Factor^S(ISD_{(i)}.A)$=0.1
	$CoSR$: $Factor^S(ISD_{(i)}.A)$=0.5
	$CrSR$: $Factor^S(ISD_{(i)}.A)$=0.9
$ISD_{(i)}.IL$	NR: $Factor^S(ISD_{(i)}.A)$=0.1
	$RBNW$: $Factor^S(ISD_{(i)}.A)$=0.5
	W: $Factor^S(ISD_{(i)}.A)$=0.9

3.3.3 加权规范化决策矩阵

由于各属性的取值主要来源于敏感信息泄露场景的客观分析及分析者的主观经验，为保证最终聚合生成敏感度的合理性，需要在聚合方法中对各种属性赋予属性权重 w，以去除主观经验对决策值的影响。面向 i 采样点的各属性值的权重表示为：$w_{ISD_{(i)}.Addr}$。鉴于存在不同类型敏感信息决策矩阵中部分属性值为 0 的情况（如对函数类敏感信息其脚本运行与身份泄露属性均为 0），因而对 3 类不同的敏感信息，根据各属性的客观重要性依次建立属性权重集为

$$\exists x^i = ISD_{(i)}, x_j^i = ISD_{(i)}.A$$

若 $ISD_{(i)}.T = Func$，则

$$\begin{aligned}\boldsymbol{w}^i &= (w_{ISD_{(i)}.Addr}, w_{ISD_{(i)}.F}, w_{ISD_{(i)}.P}, w_{ISD_{(i)}.FI}, \\ &\quad w_{ISD_{(i)}.SR}, w_{ISD_{(i)}.IL}) \\ &= (0.4, 0.2, 0.6, 0.8, 0, 0)\end{aligned} \tag{3}$$

若 $ISD_{(i)}.T = Scr$，则

$$\begin{aligned}\boldsymbol{w}^i &= (w_{ISD_{(i)}.Addr}, w_{ISD_{(i)}.F}, w_{ISD_{(i)}.P}, w_{ISD_{(i)}.FI}, \\ &\quad w_{ISD_{(i)}.SR}, w_{ISD_{(i)}.IL}) \\ &= (0, 0.3, 0.6, 0, 0.9, 0)\end{aligned} \tag{4}$$

若 $ISD_{(i)}.T = IntD$，则

$$\begin{aligned}\boldsymbol{w}^i &= (w_{ISD_{(i)}.Addr}, w_{ISD_{(i)}.F}, w_{ISD_{(i)}.P}, w_{ISD_{(i)}.FI}, \\ &\quad w_{ISD_{(i)}.SR}, w_{ISD_{(i)}.IL}) \\ &= (0.2, 0.4, 0.6, 0, 0, 0.8)\end{aligned} \tag{5}$$

3.3.4 计算理想优基点与反理想优基点

以第 4 节中抽取的语义规则为标准，为面向不同类型的敏感信息所实现的操作语义建立决策矩阵的理想优基点和反理想优基点。具体计算规则主要基于在不同操作语义中对敏感信息操作的合法属性取值区间判定，由此建立理想优基点和反理想优基点 $\boldsymbol{De}^{i^{\Delta}}$ 和 $\boldsymbol{De}^{i^{\nabla}}$。

规则 7 最安全的情况是在函数调用可见、函数名称混淆的情况下，为内部用户提供普通函数调用接口；而最不安全的情况是在函数调用可篡改、函数名称开放的情况下，为外部用户提供普通函数调用接口。

$$\boldsymbol{De}^{i^{\Delta}} = (0.325, 0.1, 0.5, 0.1, 0, 0)$$

$$\boldsymbol{De}^{i^{\nabla}} = (0.9, 0.9, 0.5, 0.9, 0, 0)$$

规则 8 最安全的情况是在函数调用可见、函数名称混淆的情况下，为内部管理员提供普通函数调用接口；而最不安全的情况是在函数调用可篡改、函数名称开放的情况下，为外部用户提供关键函数调用接口。

$$\boldsymbol{De}^{i^{\Delta}} = (0.325, 0.1, 0.1, 0.1, 0, 0)$$

$$\boldsymbol{De}^{i^{\nabla}} = (0.9, 0.9, 0.9, 0.9, 0, 0)$$

规则 9 最安全的情况是在脚本调用可见、调用接口混淆的情况下，为内部用户提供普通脚本调用接口；而最不安全的情况是在脚本调用可篡改、脚本名称开放的情况下，为外部用户提供普通脚本调用接口。

$$\boldsymbol{De}^{i^{\Delta}} = (0.325, 0.1, 0.5, 0, 0.1, 0)$$

$$\boldsymbol{De}^{i^{\nabla}} = (0.9, 0.9, 0.5, 0, 0.9, 0)$$

规则 10 最安全的情况是在脚本调用可见、调用接口混淆的情况下，为内部管理员提供普通脚本调用接口；而最不安全的情况是在脚本调用可篡改、脚本名称开放的情况下，为外部用户提供关键脚本调用接口。

$$\boldsymbol{De}^{i^{\Delta}} = (0.325, 0.1, 0.1, 0, 0.1, 0)$$

$$\boldsymbol{De}^{i^{\nabla}} = (0.9, 0.9, 0.9, 0, 0.9, 0)$$

规则 11 最安全的情况是在中间数据地址可读、数据格式为密文的情况下，为内部管理员提供

可读接口；而最不安全的情况是在中间数据可写、数据格式为明文的情况下，为外部用户提供可读接口。

$$\boldsymbol{De}^{i\Delta}=(0.325,0.1,0.1,0,0,0.5)$$

$$\boldsymbol{De}^{i\nabla}=(0.9,0.9,0.9,0,0,0.5)$$

规则 12　最安全的情况是在中间数据地址可写、数据格式为密文的情况下，为内部管理员提供可写接口；而最不安全的情况是在中间数据地址可写、数据格式为明文的情况下，为外部用户提供可写接口。

$$\boldsymbol{De}^{i\Delta}=(0.325,0.1,0.1,0,0,0.9)$$

$$\boldsymbol{De}^{i\nabla}=(0.9,0.9,0.9,0,0,0.9)$$

3.3.5　计算灰色欧几里得距离

根据理想优基点法的规则，首先计算当前敏感信息出现点 x^i 相对反理想优基点和理想优基点的欧几里得距离（M 表示敏感信息出现点的总个数）

$$\exists x^i=ISD_{(i)},x_j^i=ISD_{(i)}.A$$

生成决策集合 D 和决策权重 W

$$D_j^i=(x_1^i,\cdots,x_6^i),i\in[1,M],j\in[1,6]$$

$$w_j^i=(w_1^i,\cdots,w_6^i),i\in[1,M],j\in[1,6]$$

生成决策理想优基点及反理想优基点的欧几里得距离，定义为

$$\begin{cases}d_i^{\Delta}=\sqrt{\sum_{k=1}^{6}\left(w_k^i Factor^s(x_k^i)-De_k^{i\Delta}\right)^2}\\ d_i^{\nabla}=\sqrt{\sum_{k=1}^{6}\left(w_k^i Factor^s(x_k^i)-De_k^{i\nabla}\right)^2}\end{cases}\tag{6}$$

而后计算各敏感信息出现点处与理想优基点和反理想优基点的灰色关联系数矩阵为

$$\boldsymbol{r}_j^{i\Delta}=\frac{\min\limits_i\min\limits_j\left|De_j^{\Delta}-D_j^i\right|+\varepsilon\cdot\max\limits_i\max\limits_j\left|De_j^{\Delta}-D_j^i\right|}{\left|De_j^{\Delta}-D_j^i\right|+\varepsilon\cdot\max\limits_i\max\limits_j\left|De_j^{\Delta}-D_j^i\right|}\tag{7}$$

$$\boldsymbol{r}_j^{i\nabla}=\frac{\min\limits_i\min\limits_j\left|De_j^{\nabla}-D_j^i\right|\varepsilon\cdot\max\limits_i\max\limits_j\left|De_j^{\nabla}-D_j^i\right|}{\left|De_j^{\nabla}-D_j^i\right|+\varepsilon\cdot\max\limits_i\max\limits_j\left|De_j^{\nabla}-D_j^i\right|}\tag{8}$$

其中，ε 为分辨系数，一般取值为 0.5，而后计算灰色关联度为

$$r^{i\Delta}=\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{r}_j^{i\Delta},r^{i\nabla}=\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{r}_j^{i\nabla}\tag{9}$$

计算灰色理想优基点及灰色反理想优基点为

$$\begin{cases}s^{i\Delta}=\alpha d_i^{\nabla}+\beta r^{i\Delta}\\ s^{i\nabla}=\alpha d_i^{\Delta}+\beta r^{i\nabla}\\ \alpha+\beta=1\end{cases}\tag{10}$$

最终由灰色欧几里得距离来判决敏感信息的敏感度为

$$ISD_{(i)}.SL=C^i=\frac{s^{i\Delta}}{s^{i\Delta}+s^{i\nabla}}\tag{11}$$

敏感度阈值取决于敏感信息预泄露情况下敏感度的取值，而预泄露场景中的动态敏感度 SL 以关联系数 σ 趋近于反理想优基点，因此，可知敏感度阈值为

$$\begin{cases}\sigma=\dfrac{r^{i\nabla}}{r^{i\Delta}+r^{i\nabla}}\\ d_T^{\Delta}=\sqrt{\sum_{k=1}^{6}\left(w_k^i\cdot\sigma\cdot De_k^{i\nabla}-De_k^{i\Delta}\right)^2}\\ d_T^{\nabla}=\sqrt{\sum_{k=1}^{6}\left(w_k^i\cdot\sigma\cdot De_k^{i\nabla}-De_k^{i\nabla}\right)^2}\\ ISD_{(i)}.ST=\dfrac{d_T^{\nabla}}{d_T^{\Delta}+d_T^{\nabla}}\end{cases}\tag{12}$$

以上过程对不同属性按照相同标度进行去量纲化操作，而后计算出敏感信息流向在不同操作语义下判决矩阵的理想优基点和反理想优基点，结合属性的权重分配及灰色关联分析方法，将最终计算所得的灰色欧几里得距离作为敏感信息出现点的敏感度度量值，同时，假定操作语义为敏感信息预泄露时计算其敏感度取值，关联系数 σ 的取值来源于灰色关联度计算中反理想优基点的灰度化结果，以此赋值为敏感度阈值，评估敏感度的安全性及潜在威胁。

3.4　对比分析

为评估该方法的安全性和可用性，本文选择经典的访问控制模型 BLP 进行对比分析。BLP 基于自主访问控制和强制访问控制 2 种方式实现，使用数学语言对系统的安全性质进行描述，定义了系统、系统状态、状态间的转换规则，是第一个比较完整地对系统安全进行严格证明的数学模型，被广泛应用于描述计算机系统的安全问题。表 2 主要从模型本质、属性定义、规则定义、安全评估和可用评估几个角度对比 BLP 模型与本文所实现的方法。从模型本质看，BLP 是基于状态机来描述系统状态及状

态间的转移过程，而本文方法则是基于常规操作规则及多属性决策计算敏感度，因此，目标性有一定区分度。2 种方法均定义了访问属性、决策属性等元素，但在规则定义中，BLP 倾向于定义状态间的转移过程，而本文重在定义敏感信息的流动过程。从安全评估角度看，BLP 模型严格限制了状态机下不同安全级的状态转移规则，较安全；本文方法则聚合访问属性并参考常规操作规则聚合生成敏感信息的敏感度，安全性也较高。最后从可用性角度进行评估，BLP 模型在信息横向流动以及主客体授权方面有待改进，而本文方法若要兼容其他平台下的敏感信息安全度量，则需半自动化地收集目标信息的若干属性。

4 实验及其分析

4.1 原型系统构建

为验证该敏感度计算方法的正确性并测试泄露感知的有效性，本文方案面向大数据平台部署并实现了原型系统 TaintBigData。该系统基于 Java 编译环境对几类典型的大数据平台进行重编译，经过源代码的预处理以及词法和语法分析建立抽象的语法树，在中间代码的生成过程中抽取出控制流层的方法、脚本调用关系，用户认证中间数据生成过程，针对推演后的 3 类敏感信息集合添加污点标记[17]，结合攻击模式库为污点跟踪后的数据关联关系生成约束条件，最终给出泄露场景的感知结果。为验证场景的有效性及真实性，再进一步通过后向污点跟踪方法来生成漏洞利用验证程序（POC，proof of concept），完成已知漏洞的验证及未知漏洞的挖掘。具体流程如图 3 所示。

4.2 污点标记

原型系统中对方法（即函数）、脚本、认证过程中间数据的污点标记主要基于特征点抽取和源码重编译实现：对敏感函数调用和敏感脚本调用，在源码中抽取函数调用、脚本调用的特征代码进行污点标记；对用户认证生成的中间数据，主要抽取有数据残留的特征点，即在用户认证过程结束后该中间数据依然存在于系统内存或磁盘中。原型系统对以上特征点的抽取可实现自动化，汇聚并传递给源码重编译过程，从而为重编译后的系统提供污点跟踪支持。后期可考虑引入机器学习模型对污点标记方法进行拓展，优化污点标记

表 2 BLP 模型与本文方法的对比分析

对比项	BLP 模型	本文方法
模型本质	用来描述系统安全状态及状态转移过程	基于操作规则计算敏感信息动态敏感度
属性定义	主要包括主体、客体、访问属性和判决属性等	主要包括敏感信息、敏感信息的访问属性和判决属性
规则定义	基于状态机定义的系统状态转换规则，用于定义主体请求客体的读、添加、写、执行的访问权	基于敏感信息数据流的常规操作定义的访问规则，并总结不同操作中访问属性的可变区间
安全评估	面向状态机预定义状态的安全级，基于转换规则定义不同状态之间的信息流向，安全性较高	面向敏感信息定义特征化的访问属性，基于多属性决策聚合生成动态安全级，安全性较高
可用评估	为操作系统提供安全策略，但部门之间信息的横向流动被禁止，且缺乏灵活的授权机制	面向大数据平台中的敏感信息给出针对性的敏感度计算方法，在当前目标环境中可用性较好，若兼容其他平台，需要收集并总结相应的常规操作定义及访问属性的可变区间

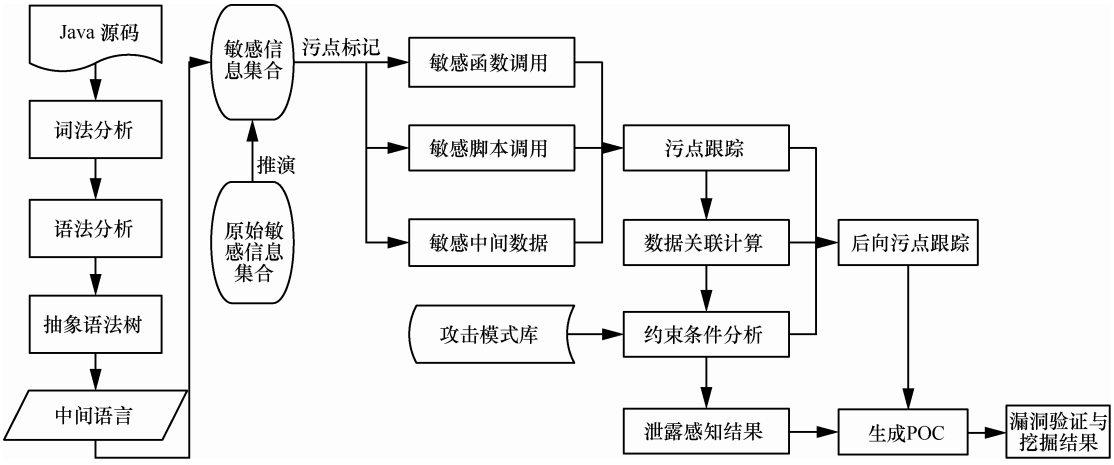


图 3 TaintBigData 设计流程

覆盖率及效率。

在大数据平台运行过程中会生成很多中间数据（主要指无法确定是否与认证过程相关的中间数据），且从用户认证的正向过程分析，暂时可确定为非目标性对象。是否标记并分析此类中间数据，需要具体参考其在内存或磁盘上的残留情况，基于正则匹配方法定时对内存及磁盘的残留数据进行扫描，若发现有认证敏感词相关的命名文件持续残留在内存或磁盘中，则根据敏感词回溯平台中相关的用户认证过程，分析生成该中间数据的输入数据约束特征。由于中间数据的安全威胁主要来源于用户认证过程中的数据泄露，因此，该过程集成在敏感中间数据的后向污点跟踪模块中实现。

4.3　实验数据分析

实验主要针对 3 类应用范围最广泛的大数据分析及处理平台 Apache Hadoop、Apache Storm 和 Apache Spark 展开，将跨平台覆盖大数据分析的离线及在线处理全过程。首先基于各平台已知敏感信息泄露漏洞建立敏感信息集，并在此基础上拓展建立面向敏感信息集的攻击模式库；然后部署本文方案所设计的原型系统，在 Java 编译环境下实现跨平台的污点插入重编译，对控制流的方法及脚本调用、数据流的认证中间数据生成过程重点标记，基于敏感信息集与攻击模式库计算污点标记的动态敏感度，对比敏感度阈值来判决是否存在敏感信息泄露场景。具体的实验环境参数如下。

1) 硬件平台

处理器：Intel(R)Core2Duo CPU E7500@ 2.13 GHz

内存：32.00 GB

显卡：NVIDIA Ge Force 9600 GSO 512

2) 软件平台

操作系统：Windows 7 旗舰版 Service Pack1

开发环境：Eclipse 3.7.1 (INDIGO)、ANTLR1.4.2、JDK1.6.0_35

运行环境：JRE1.6.0_35.b10、JVM：Hot Spot Client VM（build 20.10.b01）

目标版本：Hadoop≥2.7.0; Group Storm≥0.10.0; Spark≥1.5.0

4.3.1　目标敏感信息推演

实验基于已有漏洞对 3 种典型敏感信息集进行类型推演，对于方法（函数）类或脚本类敏感信息，根据地址、格式、权限 3 种固有属性计算相似度；对于中间数据类敏感信息，依据格式、权限等固有属性计算数据相似度。对已发现的 19 种敏感信息推演共生成敏感信息目标集合中 48 652 种目标敏感信息。此处以漏洞 CVE-2015-3188 为例，展示如何从该漏洞中抽取相关敏感信息，并完成类型推演，具体过程如图 4 所示。

如图 4 所示，针对该 CVE 漏洞下载源码补丁，通过补丁比对获得漏洞差异函数，经半自动化分析后定位核心漏洞方法（即函数）为 AddStructMetaDataMap，在静态函数调用关系图中回溯函数调用类，经过预定义的固有属性匹配检查，获得在地址、权限等属性上有相似度的方法集。由于源码数量较大，通过对补丁中获取的一项敏感信息（如函数、脚本和中间数据等）进行推演，可定位几千项敏感信息。若全部实现污点跟踪，会严重影响系统运行效率，因此，参考信息流的动态监控采用批处理对几万种敏感信息降维，对函数或脚本类敏感信息，可去除动态调用关系频繁的函数，主要对调用频率较低的方法

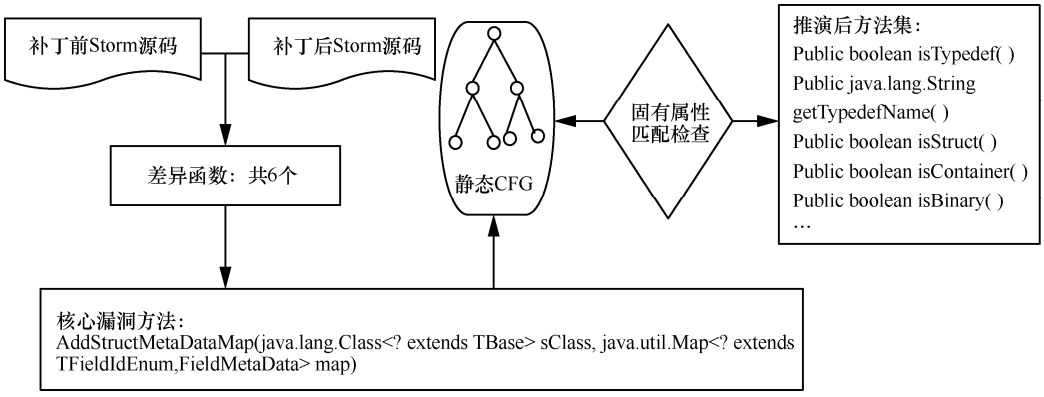
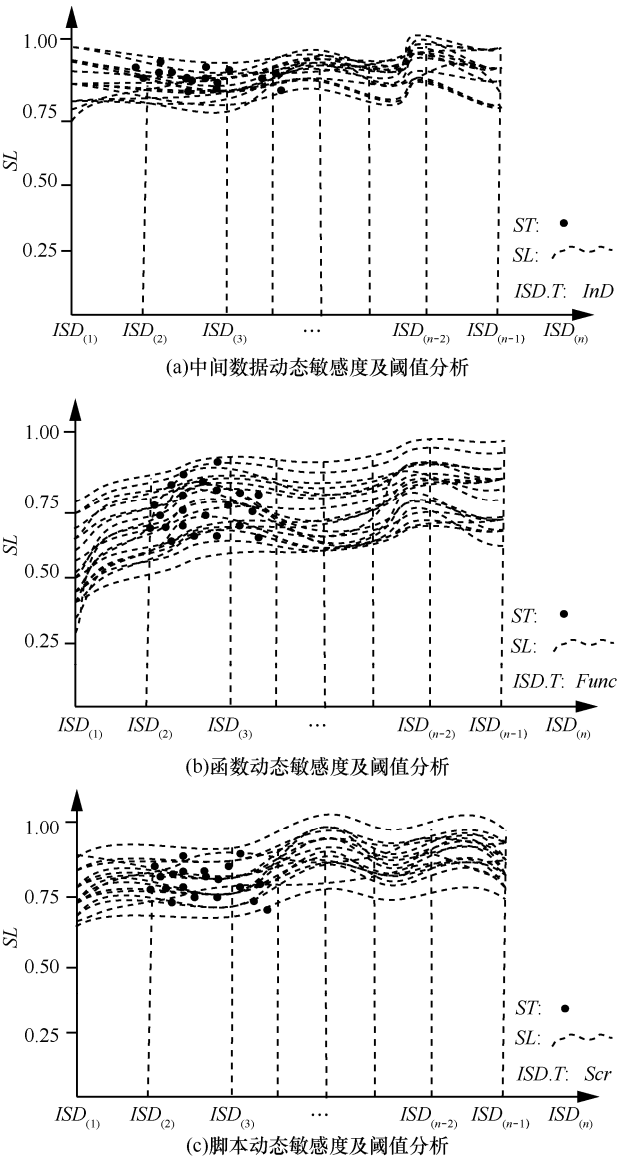


图 4　CVE-2015-3188 漏洞中目标敏感信息集合的推演

部署污点跟踪。该降维方法会导致部分漏洞漏报，但考虑系统开销后，选择对目标集合进行折中处理。降维后目标敏感信息集共包括 6 377 种敏感信息。

4.3.2 动态敏感度及敏感度阈值计算

根据之前收集所得，针对 3 种典型大数据分析平台的敏感信息泄露类漏洞进行总结，面向推演所得的敏感信息目标集合，进一步抽取出攻击模式库，主要包括方法调用的条件约束及可导致自定义方法调用的关键位置。结合目标数据集与攻击模式库，在重编译过程中为目标数据集的污点传播过程进行动态标记，并基于敏感度算法计算动态敏感度及其阈值。鉴于敏感信息集元素数量庞大，本文方案对分类后的敏感信息动态敏感度及阈值进行统计分析，具体统计数据如图 5 所示。实验对 3 类不同的敏感信息选取 20 种典型的敏感数据实例，在 n（n=100）个数据出现点中进行采样并计算动态 *SL* 以及 *ST* 的离散值，汇聚后绘制成 3 种统计数值表。首先在不同类别的敏感信息中横向对比 *SL* 以及 *ST* 的取值分布。对函数类敏感信息（如图 5(b)所示），由于函数调用所导致的任意代码执行已覆盖到系统中大量类及继承类中的若干方法，因此，总体而言敏感度取值分布较大，从 0.324 7 到 0.893 3，但通常在第 3 个采样点及第 n−1 个采样点附近出现 *SL* 的峰值，原因主要在于图 5(b)展示的 20 种敏感调用方法在这 2 个采样点处的权限较高，且调用地址可写（*ISD.P* = *OU*, *ISD.Addr* =*W*）。*ST* 值主要分布在[0.655 2, 0.811 6]区间内，在第一次出现 *SL* 峰值时，大部分 *SL* 值小于 *ST* 值，而第 2 次 *SL* 峰值有少量典型方法的 *SL* 值超过 *ST* 值。图 5(c)脚本类敏感信息（*ISD.T*=*Scr*）的 *SL* 取值相对而言比较集中，主要分布在区间[0.621 9, 0.923 7]内，在第 47 和第 95 个采样点附近出现较密集的 *SL* 峰值，整体而言 *SL* 值分布比较稳定，相比函数类波幅较小。主要原因在于任意脚本代码执行类漏洞在第 47 和第 95 个采样点处恰好是为外部用户提供的脚本调用接口，此处的地址、格式、权限属性安全性较低，因此大量典型脚本的 *SL* 动态值在此处超过 *ST* 阈值。而针对用户认证或鉴权相关的中间数据（如图 5(a)所示），*SL* 值更为集中分布，大多在[0.764 6, 0.963 2]区间内分布，*ST* 值的分布更为集中，大多在区间[0.810 4, 0.886 5]内，约一半的典型样本在第 83 个采样点处的 *SL* 值超过阈值，主要原因在于第 83 个采样点选择了大数据平台中启用认证机制后中间数据的数据销毁点，而大多数情况下平台并未能准确清除或销毁认证过程中生成的中间数据，因此该处风险较高。



(a)中间数据动态敏感度及阈值分析

(b)函数动态敏感度及阈值分析

(c)脚本动态敏感度及阈值分析

图 5　动态敏感度及阈值统计分析

4.3.3 敏感信息泄露场景判决

对目标敏感信息集中的所有敏感信息项进行 *SL* 与 *ST* 值的比较分析，共获得 6 637 种敏感信息中 452 种敏感信息在 634 个出现点处的 *ST* 值溢出点（*ISD.SL* > *ISD.ST*），根据第 5 节中的属性特征，可判决为 3 种典型的敏感信息泄露场景：对于函数类，外部用户或普通内部用户可调用某些关键函数；对于脚本类，外部用户或普通内部用户可调用某些关键脚本；对于中间数据类，外部用户或普通

内部用户可查看用于认证的中间数据明文。但经分析后发现，大量已发现的泄露场景无法直接构造漏洞利用程序实现验证过程，主要原因在于基于攻击模式库的污点跟踪以及约束条件未能准确的推演到同类敏感信息中各个敏感数据的泄露个例情况。如对函数或脚本类敏感信息，已知方法 *M* 在类 *C* 中的 *P* 位置处存在任意代码执行漏洞，在本文方案的原型系统中所抽取的攻击模式库对方法 *M'*在类 *C'*中的 *P'*位置进行 *SL* 值判决，同样确定获得类似漏洞。但实际构造漏洞利用程序时发现，*P'*处的方法调用条件与 *P* 处相比，其外部依赖关系根本不可达。因此，需要对获得的 *SL* 值溢出点进行修正，挖掘其中真正可达的敏感信息泄露点。本文方案在这里引入后向污点传播[18]的方法对 634 个敏感信息出现点构造泄露场景触发数据，分别覆盖 3 种不同种类的敏感信息对象，且分布在 3 个不同的大数据平台中，如表 3 所示，对 3 种不同的敏感信息分别进行总结，分别获得敏感信息泄露场景的个数为 5、4、7，分别覆盖 Hadoop、Storm、Spark 平台中泄露场景的个数为 8、3、6。

表 3　　敏感信息泄露场景收集

种类	Hadoop	Storm	Spark
Func	62(3)	31(1)	21(1)
Scr	43(2)	47(0)	53(2)
IntD	87(3)	64(2)	44(3)

4.3.4　已知漏洞验证及未知漏洞发现

基于敏感信息泄露场景，本文方案总结已发现的漏洞如表 4 所示，共计 17 个，包括 Hadoop 平台中的 8 个，Storm 平台中的 3 个以及 Spark 平台中的 6 个，依次命名为 Vul_A~Vul_Q。由于原型系统中使用的敏感信息集合和攻击模式库均是从已知漏洞推演而来，因此，这 17 个漏洞中包含了用于建模的 5 个已知 CVE 漏洞：CVE-2014-3627、CVE-2016-5393、CVE-2015-7430、CVE-2015-1776 和 CVE-2015-3188。如表 4 所示，分别对应到 Hadoop 及 Storm 平台中的漏洞，而 Spark 平台目前还没有公开的 CVE 漏洞号，因此，该平台相关漏洞全部为未知漏洞。由该漏洞处的 *SL* 与 *ST* 值比较可知，绝大多数漏洞发现点处的 *SL* 值明显大于 *ST* 值，且 *SL* 均值在 3 类数据集中属于偏高区间分布，可知基于动态敏感度的计算客观反映了漏洞是否存在以及漏洞所造成的敏感信息泄露风险。

表 4　　验证已知漏洞及发现未知漏洞

漏洞	平台	类型	*SL*	*ST*	结果
Vul_A	Hadoop	*Func*	0.845 1	0.471 2	未知
Vul_B	Hadoop	*Func*	0.881 3	0.485 3	2016-5393
Vul_C	Hadoop	*Func*	0.792 9	0.664 5	未知
Vul_D	Hadoop	*Scr*	0.773 2	0.539 3	2015-7430
Vul_E	Hadoop	*Scr*	0.839 5	0.611 7	未知
Vul_F	Hadoop	*IntD*	0.894 8	0.823 2	未知
Vul_G	Hadoop	*IntD*	0.880 3	0.826 3	2014-3627
Vul_H	Hadoop	*IntD*	0.911 7	0.845 9	2015-1776
Vul_I	Storm	*Func*	0.630 2	0.492 2	2015-3188
Vul_J	Storm	*IntD*	0.908 8	0.853 0	未知
Vul_K	Storm	*IntD*	0.894 0	0.829 4	未知
Vul_L	Spark	*Func*	0.695 1	0.447 3	未知
Vul_M	Spark	*Scr*	0.745 9	0.554 1	未知
Vul_N	Spark	*Scr*	0.803 5	0.609 6	未知
Vul_O	Spark	*IntD*	0.914 3	0.833 7	未知
Vul_P	Spark	*IntD*	0.905 8	0.850 1	未知
Vul_Q	Spark	*IntD*	0.883 2	0.843 9	未知

4.3.5　典型漏洞分析

限于篇幅原因，该节选择其中一个挖掘得到的未知漏洞 Vul_L 展开分析，具体属于 Spark 平台中敏感函数调用所导致的任意代码执行漏洞，存在于 Spark 2.0.1 版本中，scala 语言版本为 2.10.4。图 6 为最终的漏洞利用程序，限于该漏洞处于 CVE 号申请状态中，因此，隐去关键信息。由图 6 可知，.SetMaster 中完成对 Master 节点的参数赋值，关键点在于需要绑定该节点到 Spark-XXX-XXX-master:34XX 端口，在该处定义的 exploit 方法通过添加 *x*、*y* 两处的定义可实现对文件的下载并自动化调用。对比常规方法调用，攻击模式库中的基于元素定义的外部代码执行模式对绑定该端口处的方法成功实现漏洞利用，借助于 *x*、*y* 的定义添加下载远程目标代码（可为任意可执行的程序、脚本等），并通过平台兼容的代码执行命令将其运行。该漏洞的利用过程简便，利用结果威胁极大，截至目前，尚未出现基于此类攻击模式的恶意攻击事件，但随着大数据分析及处理平台的普及，必将出现基于此类漏洞的攻击场景。此类漏洞出现的根源在于 Spark 平台在设计

之初就假定提交任务的节点是安全的，因此，默认授予其对任意端口的访问，并忽略审计其对外部代码的调用过程。此类漏洞的修补可通过端口访问的限制策略完成，或对节点中提交的任务进行外部代码调用过程审计。

```
import org.apache.spark.{SparkContext, SparkConf}

object Exploit

  {

    def main(arg: Array[String]) {

        val sconf = new SparkConf()

            .setMaster("[redacted]")
                .setAppName("Exploit")
                .set("spark.cores.max", "12")
                .set("spark.executor.memory", "10g")
                .set("spark.driver.host","hacked.work")

        val sc = new SparkContext(sconf)

        sc.addJar("[redacted]")

        val exploit = sc.parallelize(1 to 1).map(x=>{

              val x = "[redacted]".!
              val y = "[redacted]".!
              scala.io.Source.fromFile("[redacted]").mkString
        })

        exploit.collect().foreach(println)

    }
```

图 6 Vul_L 漏洞利用程序

4.3.6 系统性能分析与比较

目前，尚未出现系统化的面向大数据分析或处理平台的安全审计或漏洞挖掘工具，也未出现大数据分析环境下成熟的数据流监督方法。为衡量原型系统的性能开销，本文方案对比了 2 种典型污点跟踪方法，主要在污点跟踪部署前后计算系统的性能损耗，并横向对比其他污点方法的性能优劣。如表 5 所示，TaintEraser[19]在函数级实现了较好的污点跟踪效果，对系统内核层以及文件访问过程的污点跟踪有突破性的效果。

表 5 性能比较

污点跟踪方案	污点部署前计算系统的性能损耗/μs	污点部署后计算系统的性能损耗/μs	平均性能损耗
TaintEraser	67.227 1	98.230 6	46.12%
TaintDroid	52.491 7	70.239 1	33.81%
TaintBigData	57.612 0	80.397 5	39.55%

该方法主要部署在二进制层，因而性能损耗较大，在污点部署后，导致 46.12%的性能损耗。而 TaintDroid[20]是面向 Android 环境的沙箱系统，但其在 Java 层有较好的污点跟踪及动态传播效果，与本方案的系统运行环境有一定的相似性。统计分析后可知其污点部署前后的性能损耗为 33.81%。而 TaintBigData 在面向 3 种大数据平台时的平均损耗达到 39.55%，相比函数级的污点跟踪工具 TaintEraser，在同时实现了应用层函数级和数据级的污点跟踪前提下也降低了性能损耗；而相比 TaintDroid，性能损耗有约 6%的提升，主要原因有如下 2 点：1) TaintBigData 并不只跟踪用户给定的敏感信息集，且需要在此基础上完成大量的目标集合元素推演；2) 为利于漏洞的验证和挖掘，TaintBigData 中通过加入了后向污点的方法来生成漏洞利用程序（POC）,该部分工作造成一定的性能损耗。因此，综合考虑系统功能的完整性及系统运行的高效性，TaintBigData 展现了较好的性能效果，损耗在可接受范围内。

5 结束语

本文面向大数据平台中已知的敏感信息泄露类型漏洞抽取并推演目标数据集，根据逆向分析中敏感信息若干特征建立多属性模型，基于灰色关联分析及理想优基点法计算敏感信息动态敏感度，通过与敏感度阈值比较来感知敏感信息的泄露场景。实验中完成了已知漏洞的验证以及若干未知漏洞的挖掘，已选择若干影响较严重的漏洞申报 CVE 号，并选择了 2 个典型的污点跟踪工具完成系统开销评估。后续需要完成目标敏感信息的自适应收集与聚类，并进一步优化动态敏感度与阈值比较结果在漏洞挖掘上的准确性，希望能直接通过敏感度算法以及污点跟踪系统来生成真实的漏洞利用程序。

参考文献：

[1] 付钰, 李洪成, 吴晓平,等. 基于大数据分析的 APT 攻击检测研究综述[J]. 通信学报, 2015,36(11):1-14.
FU Y, LI H C, WU X P, et al. Detecting APT attacks: a survey from the perspective of big data analysis[J]. Journal on Communications, 2015, 36(11):1-14.

[2] 张浩, 王丽娜, 谈诚, 等. 云环境下 APT 攻击的防御方法综述[J]. 计算机科学, 2016, 43(3):1-7.
ZHANG H, WANG L N, TAN C, et al. Review of defense methods against advanced persistent threat in cloud environment[J]. Computer Science, 2016, 43(3):1-7.

[3] 冯登国, 张敏, 李昊. 大数据安全与隐私保护[J]. 计算机学报, 2014, 37(1): 246-258.
FENG D G, ZHANG M, LI H. Big data security and privacy protection[J]. Chinese Journal of Computers, 2014,37(1): 246-258.

[4] 梁吉业, 冯晨娇, 宋鹏. 大数据相关分析综述[J].计算机学报, 2016,39(1): 1-18.
LIANG J Y, FENG C J, SONG P. A survey on correlation analysis of

big data[J]. Chinese Journal of Computers, 2016, 39(1): 1-18.

[5] 刘雅辉, 张铁赢, 靳小龙, 等. 大数据时代的个人隐私保护[J]. 计算机研究与发展, 2015, 52(1): 229-247.
LIU Y H, ZHANG T Y, JIN X L, et al. Personal privacy protection in the era of big data[J]. Journal of Computer Research and Development. 2015, 52(1): 229-247.

[6] 黄刘生, 田苗苗, 黄河. 大数据隐私保护密码技术研究综述[J]. 软件学报, 2015,26(4): 945-959.
HUANG L S, TIAN M M, HUANG H. Preserving privacy in big data: a survey from the cryptographic perspective[J]. Journal of Software, 2015, 26(4):945-959.

[7] 田洪亮, 张勇, 许信辉, 等. 可信固态硬盘: 大数据安全的新基础[J]. 计算机学报, 2016, 39(1): 154-168.
TIAN H L, ZHANG Y, XU X H, et al. Trusted SSD: new foundation for big data security[J]. Chinese Journal of Computers, 2016,39(1): 154-168.

[8] DAMGARD I, PASTRO V, SMART N, et al. Multiparty computation from somewhat homomorphic encryption[C]//Advances in Cryptology-CRYPTO 2012. 2012: 643-662.

[9] 熊金波, 张媛媛, 李凤华, 等. 云环境中数据安全去重研究进展[J]. 通信学报, 2016, 37(11): 238-250.
XIONG J B, ZHANG Y Y, LI F H, et al. Research progress on secure data deduplication in cloud[J]. Journal on Communications, 2016, 37(11): 238-250.

[10] 陈越, 李超零, 兰巨龙, 等. 基于确定/概率性文件拥有证明的机密数据安全去重方案[J]. 通信学报, 2015, 36(9): 1-12.
CHEN Y, LI C L, LAN J L, et al. Secure sensitive data deduplication schemes based on deterministic/probabilistic proof of file ownership [J]. Journal on Communications, 2015, 36(9): 1-12.

[11] CHENG H, RONG C, HWANG K, et al. Secure big data storage and sharing scheme for cloud tenants[J]. China Communications, 2015, 12(6): 106-115.

[12] 孙国梓, 董宇, 李云. 基于 CP-ABE 算法的云存储数据访问控制[J]. 通信学报, 2011, 32(7):146-152.
SUN G Z, DONG Y, LI Y. CP-ABE based data access control in cloud storage[J]. Journal on Communications. 2011, 32(7):146-152.

[13] 惠榛, 李昊, 张敏, 等. 面向医疗大数据的风险自适应的访问控制模型[J]. 通信学报,2015,36(12):190-199.
HUI Z, LI H, ZHANG M, et al. Risk-adaptive access control model for big data in healthcare[J]. Journal on Communications. 2015, 36(12): 190-199.

[14] 徐泽水. 不确定多属性决策方法及应用[M]. 北京: 清华大学出版社, 2004.
XU Z S. Uncertain multiple attribute decision making: methods and applications[M]. Beijing: Tsinghua University Press, 2004.

[15] 胡毓达. 多目标决策: 实用模型和选优方法[M]. 上海: 上海科学技术出版社, 2010.
HU Y D. Multiple target making decision[M]. ShangHai: Shanghai Scientific & Technical Publishers, 2010.

[16] 谭学瑞, 邓聚龙. 灰色关联分析:多因素统计分析新方法[J]. 统计研究, 1995, 12(3):46-48.
TAN X R, DENG J L. Grey connected analysis: a new method of multifactor statistical analysis[J]. Statistical Research, 1995, 12(3): 46-48.

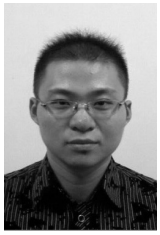
[17] 黄强, 曾庆凯. 基于信息流策略的污点传播分析及动态验证[J]. 软件学报, 2011, 22(9):2036-2048.
HUANG Q, ZENG Q K. Taint propagation analysis and dynamic verification with information flow policy[J]. Journal of Software, 2011, 22(9): 2036-2048.

[18] GANAPATHY V, JHA S, CHANDLER D, et al. Buffer overrun detection using linear programming and static analysis[C]//ACM Conference on Computer and Communications Security. 2003: 345-354.

[19] ZHU D, JUNG J, SONG D, et al. TaintEraser: protecting sensitive data leaks using application-level taint tracking[J]. ACM SIGOPS Operating Systems Review, 2011, 45(1):142-154.

[20] ENCK W, GILBERT P, HAN S, et al. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones[C]// USENIX Conference on Operating Systems Design & Implementation. 2014: 393-407.

作者简介：



沙乐天（1985-），男，江苏徐州人，博士，南京邮电大学讲师，主要研究方向为网络安全、物联网攻防等。

肖甫（1980-），男，湖南邵阳人，博士，南京邮电大学教授、博士生导师，主要研究方向为传感网和物联网等。

陈伟（1979-），男，江苏淮安人，博士，南京邮电大学教授，主要研究方向为无线网络安全、移动互联网安全。

孙晶（1985-），男，江苏宿迁人，南京电讯技术研究所工程师，主要研究方向为通信网络技术、通信技术保障。

王汝传（1943-），男，安徽合肥人，博士，南京邮电大学教授、博士生导师，主要研究方向为物联网、网络安全等。