云计算环境下隐私数据的隐式攻击保护

陶林波[1] 沈建京[1] 薛 猛[1] 蔡立刚[2]
(解放军信息工程大学理学院 郑州 450001)[1] (沈阳工业大学理学院 沈阳 110870)[2]

摘 要 云计算环境下用户的隐私数据保护成为了云计算能否快速发展的关键,目前更多的研究集中在了隐私数据的加密等显式保护方法上,而对集中存储带来的统计分析以及数据挖掘等隐式攻击带来的隐私泄露关注不够。针对云计算环境下用户隐私数据面临的数据挖掘带来的隐私泄露风险,建立了一种保护模型。其将数据在云端的录入、存储、还原、销毁几个生命周期的关键环节作为保护目标,建立相应的保护模块,对隐私数据进行相应的处理。最后通过复杂度对比的方法证明了本文模型的有效性。

关键词 云计算,数据挖掘,隐私保护,隐式攻击
中图法分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.036

Privacy Protection from Implicit Attacks in Cloud Computing Environment

TAO Lin-bo[1] SHEN Jian-jing[1] XUE Meng[1] CAI Li-gang[2]
(School of Sciences, PLA Information Engineering University, Zhengzhou 450001, China)[1]
(School of Sciences, Shenyang University of Technology, Shenyang 110870, China)[2]

Abstract Privacy protection in cloud computing environment has become the key to the rapid development of cloud computing, until now more researches focus on the explicit protection methods of privacy protection such as encryption, lacking of enough attention to implicit attack, such as data mining attack. A protection model was established in this paper according to how to protect privacy in cloud computing environment against data mining attack. The model selects several key objectives as the key protection during the life cycle of user's data, including input, storage, reduction and destruction. Then correlative modules were established according to the key objectives corresponding to data privacy. Finally, the effectiveness of this model was proved through the complexity comparison.

Keywords Cloud computing, Data mining, Privacy protection, Implicit attack

随着信息的爆炸式增长,不论是个人数据还是企业数据都呈现一种高速增长的态势。在信息共享带来便捷的同时,数据管理的成本也在不断增加,为了切实降低成本,做到按需使用,云计算受到了越来越多的关注。云计算是融合了分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡、热备份冗余等传统计算机和网络技术的发展产物,它不是一门具体技术,而是一种服务[1,2]。云计算通过将海量数据存储在分布式计算机上,极大地降低了数据的运行及管理成本,用户只需要按照所使用的实际计算及存储量进行付费即可,实现了按需使用[3]。同时,数据的集中管理也为多项服务的联合数据挖掘提供了条件[4]。在这个过程中,用户最关心的就是隐私安全问题,云计算的架构及数据的管理过程对用户来说是透明的,用户不知道自己的数据存储在哪里,以哪种形式存储,有几个副本,同时也不知道本身的隐私信息是否被泄露或滥用[5,6]。尽管云服务商承诺遵守相关的规则,但不可避免地会产生内部攻击和外部攻击,尤其是数据长期积累带来的规律性分析。云计算的最终目标就是为用户提供更加廉价、个性化的服务,廉价通过海量数据的分布式存储及数据中心的集中管理来实现;而个性化的服务则主要依靠数据挖掘来发现大量用户数据中的固定模式[7],从而为用户提供针对性强的服务。在这个过程中,用户的隐私泄露往往被忽视,美国的棱镜事件实际上就是通过长期搜集用户数据,基于数据挖掘技术发现用户的生活、学习、消费等习惯以及一些用户的隐私信息,这势必对云计算的信任度造成不良的影响。在隐私保护方面,人们考虑更多的是显式的信息篡改保护,而不是隐式的隐私挖掘保护,而后者往往给用户隐私带来全面化、系统化的威胁,同时这方面的威胁又是难以溯源的。本文针对数据挖掘产生的隐式攻击造成的隐私泄露,从数据的录入、存储、还原、销毁几个环节入手,设计了一种隐式攻击下的隐私保护模型。模型对隐私数据在这几个关键环节下的数据形式进行了处理,使用户的隐私数据难以被发现,同时又不影响正常数据挖掘的使用,在提高了用户体验的同时,保证了用户数据的安全性与效率性。

1 云计算环境下的数据挖掘与隐私泄露

本节将介绍由数据挖掘对云数据产生的隐式攻击的相关概念和理论。

到稿日期:2015-08-16 返修日期:2015-12-08

陶林波(1979—),男,博士生,工程师,主要研究方向为网络安全、大数据挖掘,E-mail:taotlb@126.com;沈建京(1961—),男,博士,教授,主要研究方向为人工智能、云计算;薛 猛(1991—),男,硕士生,主要研究方向为云计算;蔡立刚(1979—),男,硕士,讲师,主要研究方向为运筹学与控制论。

云端用户交互一般都是基于一个 Web 客户端，或一组 Web 服务，因此大多数威胁都留给了服务商[8]。而事实上，服务商虽然能够处理大部分的安全问题，但这并不意味着用户数据是绝对安全的，服务商为了提高用户的交互体验，往往要对用户数据进行针对性的挖掘，这个过程要将用户数据经过逻辑处理与关联再返回到用户交互界面。这个处理过程不再是简单的数据片段拼接，而是完整的信息生成过程。为了保证用户隐私数据的安全性，通常服务商更多地关注于显式威胁，也采取了更多的防范手段。但现在越来越多的用户隐私泄露来自于隐式攻击，因为为了方便共享和访问以及提高交互性等，用户数据的分布式存储、冗余备份中很多数据以明文呈现，这对于有目的的隐私挖掘是很方便的[9]。

云计算这种数据管理模式造成的数据高度集中与委托式管理恰恰为信息挖掘提供了条件。数据挖掘一般是指从大量的数据中通过算法搜索隐藏于其中的信息的过程[10,11]。数据挖掘的条件就是并行计算、海量数据以及人工智能技术。而云计算为数据挖掘提供了一个很好的平台——一种分布式的海量数据存储[12]。特别是以云计算为平台的大数据挖掘、联合数据挖掘，极好地发现了用户消费模式、行为模式，为用户提供了更加个性化的服务，做到服务具有预判性和针对性。这个过程中除了有益于提高用户体验的有用信息被挖掘出来的可能性外，有些敏感信息也被挖掘出来，而这些信息并不是用户愿意被共享或商业利用的。

综上，可以将云计算环境下的隐私威胁形式化表示为：

$$threat=\lambda_1 threat_1+\lambda_2 threat_2 \tag{1}$$

其中，$threat$ 为用户数据在云计算环境下的隐私威胁，$threat_1$ 为显式威胁，$threat_2$ 为隐式威胁，λ_1 和 λ_2 为 $threat_1$ 和 $threat_2$ 的权重比。近几年，用户隐私威胁中隐式威胁的权重正在不断增加。

针对云中的隐私信息，通常的保护方法是数据加密，但这种方法在提高安全性的同时降低了服务的可用性，因为每一次读取数据都要解密，每一次数据更新都要重新加密，这势必会降低数据的读取效率。如果能在当前云端数据管理的基础上只针对一小部分隐私数据进行处理，将会极大地提高安全性与用户体验。

2 云计算环境下的隐私保护机制设计

本节以数据在云计算环境下的生命周期为主线，设计一种隐私数据在数据挖掘这种隐式攻击下的保护方法。

2.1 云计算环境下数据的隐式攻击保护模型

在用户将数据提交到云端时，用户与云服务商实际上形成了一种信任的委托关系，即用户认为云服务商是可信的，同时云服务商为用户数据提供最大化的安全保证。这个过程中，数据的管理模式从传统的本地管理变成了云端管理，也就是可信的第三方管理[13]。由于数据脱离了数据所有者的管理，云服务商成为了数据的实际管理者，在这个过程中若要切实保证用户的隐私安全，又不影响数据的处理效率，就要对数据进行合理分类，并对其中比例较小的关键数据在几个关键环节实施有效保护，这样才能够达到有效保护用户隐私的目的。本文将数据的输入、存储、销毁几个环节作为数据在云中保护的关键环节，设计了一种隐式攻击下的隐私保护模型。模型结构如图 1 所示。

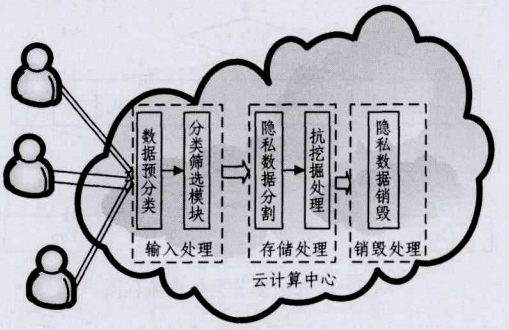

图 1 数据安全处理架构图

2.2 数据的输入处理

用户的所有数据都要经过录入、存储、挖掘分析、信息销毁等步骤，原始录入数据是后续步骤的基础，后续步骤都是在其上进行处理的。合理的数据分类将为后续处理带来极大的益处。

根据图 1 的模型结构，输入处理部分要求用户对输入的数据进行预先分类说明，对不允许共享或挖掘的隐私数据进行标注，从而达到对输入数据的前期筛选。实现这个过程的方法就是在用户输入数据时，在对应的数据输入项中都添加相应的说明或选项，使用户对自己的输入数据归属类有一定的了解。如果用户明确选择了隐私信息，则表示该数据项不允许共享或关联到其他信息。如果用户未选择，并不表示该数据项可以共享，因为存在着用户对分类说明没有正确理解或忽略了相应的安全提示的可能。所以数据在经过预分类模块之后，还要经过分类筛选模块进行进一步的筛选。

过程具体描述为：首先将原始数据划分为禁止挖掘数据集与可挖掘数据集两类，分别记为 Dm_1 与 Dm_2。设分类标识位为 pl，$pl\in\{0,1\}$，对任意输入项 d_i，若有 $d_i\in Dm_1$，则 $pl=0$，表示用户不允许公开或共享该数据项，这样即使在数据挖掘条件下也不会被他人得到关联信息；如果 $d_j\in Dm_2$，则 $pl=1$，表示用户允许共享或在用户授权或非授权情况下经过数据挖掘得到的关联信息对用户隐私都构不成严重威胁。我们知道合理的数据挖掘是有益的，它可以发现用户很多潜在的关联习惯，为用户的日常生活带来益处，所以本文主要研究 Dm_1 中数据的保护方法。

对于要存储到云端的数据输入序列 $\sigma_i=d_{i1}\,d_{i2}\cdots d_{ik}$，存在若干子项 $d_{im},\cdots,d_{in}$，有 $pl=0$，则将这些子项 $d_{im},\cdots,d_{in}$ 划分到 Dm_1 中。对于序列 σ_i 中除去子项 $d_{im},\cdots,d_{in}$ 后的剩余子项 $d_{ip},\cdots,d_{iq}$，有 $pl=1$，但并不把这些数据直接划入 Dm_2，而是等待筛选模块的进一步判断。至此，输入数据的预分类已经完成。为了保障用户隐私信息的完整性与保护的智能化，需要对 $pl=1$ 的所有子项进行进一步的判断，在筛选模块中通过内置的数据挖掘算法对 $pl=1$ 的子项进行自动筛选，通过对大量用户隐私数据的模式分析，提取一些关键隐私数据模式作为筛选判断的标准，再采取保留最小子集的策略，只保留核心隐私模式作为判断标准，从而降低筛选模块对数据挖掘结果的影响，同时做到精确保护关键隐私。输入数据分类处理过程如图 2 所示。

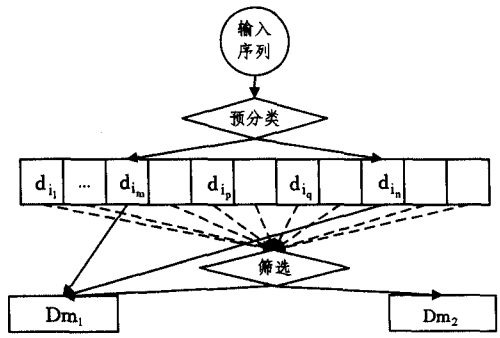

图 2　输入数据分类处理流程图

如图 2 所示，$pl=1$ 的子项在经过进一步筛选后如果包括满足筛选条件的子项，则将这些子项的 pl 值由 1 改为 0，并将这部分子项归入 Dm_1，其余子项的 pl 值不变，归入 Dm_2。经过上述处理后，数据的输入处理结束。

2.3　隐私数据的存储处理

该部分设计的目的就是对用户的隐私数据进行处理，即使用户很少参与管理数据，但是用户的隐私数据仍然由用户实际掌握；在将数据上传至云端后，云服务商得到的隐私数据对他人是不可知的，或是变换形式的，即使云服务商也不能实际掌握这些隐私数据的实际内容，他们掌握的只是隐私数据变化后的形式数据。

经过前面输入处理的数据首先实现了分类处理，也是本文模型的第一步；接下来要对 Dm_1 中的数据进行进一步的抗挖掘保护处理，对于 Dm_2 中的数据，本文不做特殊处理，以便提高数据的处理效率。在处理之前，需要补充两个定理。

定理 1　Dm_1，Dm_2 为有限集，且 Dm_1 的势小于 Dm_2 的势。

证明：对于 Dm_1 中的任意子项 d_{1i}，总可以通过自然数对其进行编号，所以有 $|Dm_1|\leqslant n$；又因为 Dm_1 的子项为单个用户关联的隐私数据，不可能无限扩展，所以必然有 $|Dm_1|<n$，故 Dm_1 为有限集；同理可证 $|Dm_2|<n$，所以 Dm_1，Dm_2 都为有限集；对于初始的 Dm_1，Dm_2，有 $Dm_1=\varnothing$，$Dm_2=\varnothing$，用户数据的扩展过程就是 Dm_1，Dm_2 集的扩大过程，由于 Dm_1 为隐私数据集，其扩展速度远低于普通数据的扩展速度，因此有 $|Dm_1|<|Dm_2|$。

定理 2　Dm_1 中的数据项构成了用户隐私空间的一组基，用户的隐私信息都可以由这组基线性表示。

证明：Dm_1 中存储的是用户的一些基本隐私数据，如身份证号、银行卡号等信息。将这些基本信息项设为 $item_1$，$item_2$，…，$item_n$。这 n 个基本的数据项中的任意一项无法用其余的 $n-1$ 项线性表示，所以这 n 项基本数据项是线性无关的，可以作为 Dm_1 的一组基。对于用户的任意一个隐私信息 $provicy_i$，其总是可以表示为 $provicy_i=\alpha_1 item_1+\alpha_2 item_2+\cdots+\alpha_n item_n$，$\alpha_i\in\{0,1\}$。所以用户的隐私信息可以由 Dm_1 的这组基线性表示。

由定理 1、定理 2 可知 Dm_1 是用户隐私集的最小子集，所以只要将 Dm_1 中的子项进行抗挖掘处理，用户的最关键隐私数据就不会外泄。本环节将通过两个安全策略提高 Dm_1 数据集的保护强度。

策略 1　拆分 Dm_1 为两个子集，对子项做逆序处理。

对于一个完整的 Dm_1 数据集，总是可以找到相应的方法还原里面的完整数据。如果将 Dm_1 拆分成两个部分并分开保存，不论是还原 Dm_1 的难度还是 Dm_1 中数据的抗挖掘能力都将显著提高，就好像一份私密文件被分解成两部分并由两个不同的人保管，即使小偷拿到了一半，没有另一半，他也没办法知道里面完整的内容。这样做的优点是增加了还原数据的难度，同时通过两个部分的相互检查保证了隐私数据的完整性，即有效地防御了外部攻击，也有效地防御了内部攻击。

处理过程如下：将 Dm_1 拆分成两个子集 $subDm_1'$ 和 $subDm_1''$，每一个 $pl=0$ 的子项 $item_i$ 也会被拆分成两个相应的子串 $subitem_i'$ 和 $subitem_i''$。为了建立两个子串的关联，增加一个 $itemid_i$ 作为 $subitem_i'$ 和 $subitem_i''$ 在 Dm_1 中的序号。这样，一个 $pl=0$ 的子项 $item_i$ 便形成了两个向量 $v_i'=(id,itemid_i,subitem_i',cf_i,dFlag_i')$，$v_i''=(id,itemid_i,subitem_i'',cf_i,dFlag_i'')$，其中 id 为用户在云中的 ID，也是云识别不同用户的依据，是用户身份的唯一标识；cf_i 为子串 $item_i$ 被拆分的位置；$dFlag_i'$ 为 $subitem_i'$ 的销毁标志位；$dFlag_i''$ 为 $subitem_i''$ 的销毁标志位，表示该子串是否已失效，如果失效则做销毁处理。为了保证分割之后的两个子项在得到 $itemid_i$ 的情况下不会被直接还原，本模型将对经过拆分后的两个子串做对应的逆序处理，处理过程为：将向量 $v_i'=(id,itemid_i,subitem_i',cf_i,dFlag_i')$ 改为 $v_i'=(id,itemid_i,subitem_i',cf_i,tp_i',dFlag_i')$，将 $v_i''=(id,itemid_i,subitem_i'',cf_i,dFlag_i'')$ 改为 $v_i''=(id,itemid_i,subitem_i'',cf_i,tp_i'',dFlag_i'')$，$tp_i'$ 为 $subitem_i'$ 的逆序起始位置，tp_i'' 为 $subitem_i''$ 的逆序起始位置，对子串 $subitem_i'$ 中在 tp_i' 后的数据做逆序处理，同理对 $subitem_i''$ 中在 tp_i'' 后的数据做逆序处理。

隐私数据存储处理结构图如图 3 所示。

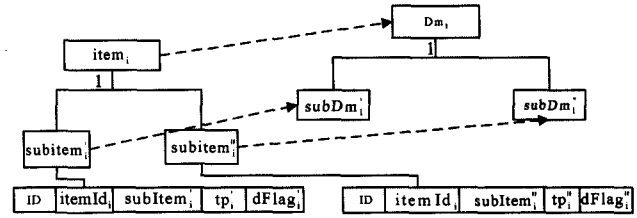

图 3　隐私数据存储处理结构图

子串的逆序处理如图 4 所示。

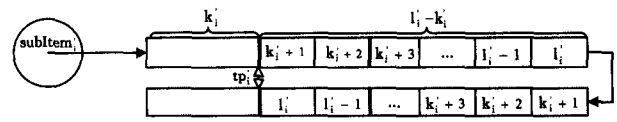

图 4　隐私数据片段处理图

图 4 给出了子串 $subitem_i'$ 的处理流程，子串 $subitem_i''$ 的处理同理。子串的还原过程则是图 4 的逆过程，只要找到对应的 tp_i'，tp_i''，再把后面的数据逆序即可。

策略 2　合并重复数据，减少写入与读取隐私数据项的次数。

很多关联性的隐私泄露都是积累性产生的，为此对于 Dm_1 中的数据项，多次输入的则进行合并；对于 Dm_2 中的数据项，由于其设定为允许挖掘空间，因此很多风险是预先不可

判断的，所以此处要进行必要的判断：若 Dm_2 中的数据项出现了可由 Dm_1 的基线性表示的项，则直接将该项数据的出现频率降为 1，随机保留一处该数据项，删除其他位置的该信息项，从而大幅削减其关联隐私信息的概率。

2.4 隐私数据的销毁

通过前面的处理，隐私数据在录入及存储环节的安全性有所保证，接下来讨论隐私数据在销毁过程中的安全处理。

$subitem_i'$ 和 $subitem_i''$ 的销毁标志位分别为 $dFlag_i'$ 和 $dFlag_i''$，默认数据子项在建立时有 $dFlag_i'=0$，$dFlag_i''=0$，如果要将数据项 $item_i$ 销毁，则需要将其两个子项分别销毁。隐私数据的销毁存在着两种情况，即正常销毁与异常销毁，此处将分别进行讨论。

正常情况下的隐私数据项销毁，只要通过 $itemid_i$ 找到对应的两个子项 $subitem_i'$ 和 $subitem_i''$，然后分别将 $dFlag_i'=0$ 和 $dFlag_i''=0$ 置为 $dFlag_i'=1$ 和 $dFlag_i''=1$，对子项 $subitem_i'$ 和 $subitem_i''$ 的值进行随机数填充，就能完成正常情况下该隐私数据项的销毁。异常情况下的隐私数据项销毁通常指销毁标志位 $dFlag_i'$ 和 $dFlag_i''$ 没有及时置为 1，或因服务商的原因造成在没有经过用户授权情况下数据部分残缺。本文要做的就是对这些残缺数据进行销毁，判断的标准则是数据的完整性，如果 $dFlag_i'$ 和 $dFlag_i''$ 的值不一致，则说明销毁过程出现异常，将两者的值全部置为 1，并进行随机数填充，从而完成销毁。如果 $dFlag_i'=0$，$dFlag_i''=0$，但 cf_i 的值不一致，说明有一部分数据已损坏，则置 $dFlag_i'=1$，$dFlag_i''=1$，然后进行随机数填充。如果对任意一个 $subitem_i'$ 或 $subitem_i''$ 无法找到对应子项，同样说明该数据项已经部分损坏，执行同上的操作。

3 模型的复杂度分析与效能评估

前面已经对隐私数据项在云计算环境下生命周期中的不同阶段进行了对应处理，接下来就要证明该模型的有效性。通常证明一个模型的有效性往往从实验数据对比或理论的复杂度比较来进行说明[14]，本文将从复杂度的证明入手来说明该模型有效地提高了隐私数据在云环境中的抗数据挖掘能力。

实验中，前、后两半子项结构表的信息如表 1、表 2 所列。

表 1 前半子项结构表

用户 ID	数据编号	前半子串值	变换起始位	删除标志位
ID	$itemid_1$	$subitem_1'$	tp_1'	$dflag_1'$
ID	$itemid_2$	$subitem_2'$	tp_2'	$dflag_2'$
ID	…	…	…	…
ID	$itemid_n$	$subitem_n'$	tp_n'	$dflag_n'$

表 2 后半子项结构表

用户 ID	数据编号	后半子串值	变换起始位	删除标志位
ID	$itemid_1$	$subitem_1''$	tp_1''	$dflag_1''$
ID	$itemid_2$	$subitem_2''$	tp_2''	$dflag_2''$
ID	…	…	…	…
ID	$itemid_n$	$subitem_n''$	tp_n''	$dflag_n''$

由表 1、表 2 可知，在用户正常读取 Dm_1 中的数据时，通过 ID 查找子空间 $subDm_1'$ 和 $subDm_1''$ 中对应的 $itemid_i$ 值，通过 $itemid_i$ 值找到对应的 $subitem_i'$ 和 $subitem_i''$ 的值，再查找 $dflag_i'$ 和 $dflag_i''$ 的值，如果其中一个为 1，则说明该数据为删除数据，已失效，应该删除，则将对应 $subitem_i'$ 和 $subitem_i''$ 的值进行随机数填充处理，并将对应删除标志位置为 1；如果删除标志位都为 0，则说明数据有效，再查找对应 tp_i' 和 tp_i'' 的值，对 $subitem_i'$ 和 $subitem_i''$ 的值进行还原处理，将其合并后返回给用户。这个过程中还原 $subitem_i'$ 和 $subitem_i''$ 值的复杂度最大，分别为 n_i' 和 n_i''，它们分别代表 $subitem_i'$ 和 $subitem_i''$ 的长度。所以该过程的算法复杂度为 $n_i'+n_i''$。如果要在数据挖掘的统计分析条件下还原该隐私数据，需要对 n_i' 和 n_i'' 长度的子串进行全排列，复杂度分别为 $n_i'!$ 和 $n_i''!$，则该过程需要的算法复杂度为 $n_i'!+n_i''!$，所以该模型对于隐私数据在云环境中抵御数据挖掘方法造成的隐私泄露具有一定的效果。

结束语 本文以数据在云中的生命周期为主线，对输入、存储、销毁几个关键环节设计了相应的处理模块，既不影响访问效率，同时又能避免敏感数据被挖掘的风险。最后通过算法复杂度分析说明了模型的有效性。

参 考 文 献

[1] Lin Wei-wei, Qi De-yu. Survey of Resource Scheduling in Cloud Computing [J]. Computer Science, 2012, 39(10): 1-6(in Chinese)
林伟伟，齐德昱. 云计算资源调度研究综述[J]. 计算机科学，2012，39(10)：1-6

[2] Hou Hong-feng, Wang Can, Wang Li-juan. Research onInformation Security Issues of Cloud Computing [J]. Journal of Electronic Design Engineering, 2012, 20(22): 60-62(in Chinese)
侯洪凤，王璨，王立娟. 云计算信息安全问题探讨[J]. 电子设计与工程，2012，20(22)：60-62

[3] Chowdhury A, Tripathi P. Enhancing Cloud Computing Reliability Using Efficient Scheduling by Providing Reliability as a Service[C]// 2014 International Conference on Parallel, Distributed and Grid Computing. 2014: 99-104

[4] Hua Bei, Zhong Cheng. The Privacy Preserving Technology in Data Mining[J]. Microelectronics & Computer, 2009(8): 26-28 (in Chinese)
华蓓，钟诚. 数据挖掘中的隐私保护技术进展分析[J]. 微电子学与计算机，2009(8)：26-28

[5] Marozzo F, Talia D, Trunfio P. A Cloud Framework for Parameter Sweeping Data Mining Applications[C]// 2011 Third IEEE International Conference on Cloud Computing Technology and Science. 2011: 367-374

[6] Yang Hang, Fong S. Countering the Concept-drift Problem in Big Data Using iOVFDT[C]// 2013 IEEE International Congress on Big Data. 2013: 126-132

[7] Ding Jing, Yang Shan-lin, Luo He, et al. Data Mining Service Model in Cloud Computer Environment[J]. Computer Science, 2012, 39(6A): 217-220(in Chinese)
丁静，杨善林，罗贺，等. 云计算环境下的数据挖掘服务模式[J]. 计算机科学，2012，39(6A)：217-220

[8] Wang Zhen-qi, Li Hai-long. Research of Massive Web Log Data Mining Based on Cloud Computing [C] // 2013 International Conference on Computational and Information Sciences. 2013: 591-594

（下转第 196 页）

表 6 第三方应用漏洞检测结果

样本	公开组件数/总组件数	Sink 类型	组件内/组件间	分析时间(s)
Sample1	11/58	开启网页	组件内	155
Sample2	5/37	开启网页(可执行远程命令)	组件内	65
Sample3	15/72	修改数据库	组件内	192
Sample4	9/63	执行命令	组件内	131
Sample5	13/51	开启网页	组件间	195
Sample6	8/49	修改数据库	组件内	55
Sample7	10/57	向指定号码发送短信	组件内	320
Sample8	7/44	写入指定文件	组件内	116

结束语 本文提出了一种基于静态污点分析的 Android 应用 Intent 注入漏洞检测方法,设计并实现了原型工具 IntentTracer。本方法同时考虑组件内和组件间的污点数据跟踪,可以有效减少误报和漏报;另外,组件分析只关注来自应用外部的 Intent 消息处理所经过的代码段而非盲目地分析所有应用组件,有效降低了性能开销。

目前的分析方法并没有考虑程序代码中的反射机制和 JNI 调用情况,因此静态分析结果存在一定误差,对于这类问题,今后将做进一步研究。

参 考 文 献

[1] Maji A K, Arshad F, Bagchi S, et al. An empirical study of the robustness of inter-component communication in Android[C]// 2012 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2012: 1-12

[2] Sasnauskas R, Regehr J. Intent fuzzer: crafting intents of death [C]// Proceedings of the 2014 Joint International Workshop on Dynamic Analysis (WODA) and Software and System Performance Testing, Debugging, and Analytics (PERTEA). ACM, 2014: 1-5

[3] Chin E, Felt A P, Greenwood K, et al. Analyzing inter-application communication in Android[C]// Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM, 2011: 239-252

[4] Lu L, Li Z, Wu Z, et al. Chex: statically vetting android apps for component hijacking vulnerabilities [C] // Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM, 2012: 229-240

[5] Octeau D, McDaniel P, Jha S, et al. Effective inter-component communication mapping in android with epicc: An essential step towards holistic security analysis[C]// USENIX Security 2013. 2013: 543-558

[6] Sagiv M, Reps T, Horwitz S. Precise interprocedural data flow analysis with applications to constant propagation[J]. Theoretical Computer Science, 1996, 167(1): 131-170

[7] Gallingani D, Gjomemo R, Venkatakrishnan V N, et al. Static detection and automatic exploitation of intent message vulnerabilities in Android applications[OL]. http://www.ieee-security.org/TC/spw2015/Most/papers/s3p1.pdf

[8] Enck W, Octeau D, McDaniel P, et al. A Study of Android Application Security[OL]. http://www.usenix.org/legacy/events/secll/tech/full_papers/Enck.pdf

[9] Takeshi Terada/Mitsui Bussan Secure Directions, Inc. Attacking Android browsers via intent scheme URLs[OL]. http://www.mbsd.jp/whitepaper/InterScheme.pdf

[10] Wang R, Xing L, Wang X F, et al. Unauthorized origin crossing on mobile platforms: Threats and mitigation[C]// Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. ACM, 2013: 635-646

[11] Lam P, Bodden E, Lhoták O, et al. The Soot framework for Java program analysis: a retrospective[C]// Cetus Users and Compiler Infastructure Workshop (CETUS 2011). 2011

[12] Bartel A, Klein J, Le Traon Y, et al. Dexpler: converting android dalvikbytecode to jimple for static analysis with soot[C]// Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program Analysis. ACM, 2012: 27-38

[13] Luo T, Hao H, Du W, et al. Attacks on WebView in the Android system[C]// Proceedings of the 27th Annual Computer Security Applications Conference. ACM, 2011: 343-352

(上接第 187 页)

[9] Dev H, Sen T, Basak M, et al. An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks[C]// 2012 SC Companion: High Performance Computing, Networking Storage and Analysis. 2012: 1106-1115

[10] Zheng Xian. Status and Development of Web Mining[J]. Journal of Technology and Market, 2013, 20(3): 64-65(in Chinese)
郑弦. Web 挖掘的现状和展望[J]. 技术研发, 2013, 20(3): 64-65

[11] Li Ling-juan, Zheng Shao-fei. Analysis of Data Mining Privacy Preserving Technology Based on Data Processing [J]. Journal of Computer Technology and Development, 2011, 21(3): 94-97(in Chinese)
李玲娟, 郑少飞. 基于数据处理的数据挖掘隐私保护技术分析[J]. 计算机技术与发展, 2011, 21(3): 94-97

[12] He Yao, Wang Wen-qing, Xue Fei. Study of Massive Data Mining Based on Cloud Computing[J]. Computer Technology and Development, 2013, 23(2): 69-72(in Chinese)
贺瑶, 王文庆, 薛飞. 基于云计算的海量数据挖掘研究[J]. 计算机技术与发展, 2013, 23(2): 69-72

[13] Ding Yan, Yang Qing-ping, Qian Yu-ming. Architecture and Key Technology of a Data Mining Platform Based on Cloud Computing[J]. ZTE Technology Journa, 2013, 19(1): 53-57(in Chinese)
丁岩, 杨庆平, 钱煜明. 基于云计算的数据挖掘平台架构及其关键技术研究[J]. 中兴通信技术, 2013, 19(1): 53-57

[14] Abuhussein A, Bedi H, Shiva S. Evaluating Security and Privacy in Cloud Computing Services: A Stakeholder's Perspective[C]// The 7th International Conference for Internet Technology and Secured Transactions. 2012: 388-395