基于弱水印的地理数据拓扑完整性检验方法

隋莉莉 汪传建
(石河子大学信息科学与技术学院 石河子 832003)

摘 要 拓扑关系是空间关系研究的基础问题,被广泛应用在空间查询和空间推理等方面。为了检测地理数据拓扑关系的完整性,提出了一种基于弱水印的地理数据拓扑完整性检验方法。通过地物之间空间相离距离生成水印信息,并修改其相离距离比值,根据比值进行地物缩放以达到嵌入水印的目的。在水印检测时,根据生成的水印与提取的水印的匹配结果来判定地理数据的拓扑完整性。实验结果表明,本方法可以有效地检测出矢量地理数据的拓扑完整性。

关键词 弱水印,矢量地图,拓扑完整性,篡改检测
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.2.040

Checking Topological Integrity of Geographic Data Based on Fragile Watermarking

SUI Li-li WANG Chuan-jian
(School of Information Science and Technology, Shihezi University, Shihezi 832003, China)

Abstract Topological relation is the basis of spatial relationships, which is widely used in spatial query and spatial reasoning. In this paper, a fragile watermarking method was proposed to verify the topological integrity of geographic data. The watermark is generated from the disjoint distance between the objects. Then the objects are zoomed in or out according to the modified disjoint distance rate based on the watermark to embed the watermark. In watermarking detection, the topological integrity of geographic data is determined by the result of matching the regenerated watermark with the extracted watermark. Experimental results demonstrate that it is effective to verify topological integrity of geographic data by the method.

Keywords Fragile watermarking, Vector map, Topological integrity, Tempering detection

1 引言

如今随着互联网的高速发展,数据的传递与交换已经变得非常便捷,人们可以方便地使用各类发布在第三方平台上的数据。但与此同时,如何保护数字产品的版权、如何验证数据内容的完整性以及如何保障数据的拥有者和使用者的权益等是所面临的问题。水印作为一种信息隐藏技术,可以较好地满足对数据的版权保护和内容完整性验证的要求。数字水印,包括鲁棒水印和脆弱水印,鲁棒水印主要用于数据的版权保护,脆弱水印主要用于数据的真伪辨别和完整性鉴定。所谓脆弱水印,就是在保证一定数据质量的前提下,将数字水印嵌入到数据中,当数据内容发生变化或被篡改时,通过提取嵌入的水印信息来鉴别数据内容的真伪,并将指出数据的篡改和变化位置甚至是攻击类型等[1]。

在地理信息领域中,矢量数字地图具有高精度、易存储和缩放不失真等优点。为了满足数据的生产者和使用者对所发布的矢量数字地图的版权和内容完整的需求,许多学者已经将水印应用于矢量数字地图中,以达到识别版权、验证完整性的目的。在验证数据完整性方面,通过无损压缩和数字摘要的方式将生成的水印信息在地理数据的误差范围内替换原坐标值,这样可以准确地定位被篡改的顶点[3-5]。文献[6,7]分别用量化和映射的思想将水印嵌入到数据中,以检测噪音和裁剪篡改。而文献[8]分两次将水印分别嵌入到顶点坐标的最低有效位和次低有效位,从而可以定位篡改位置并判断篡改类型。此外,通过对特征点的分组将可逆水印以坐标点的方式插入到每组中,这样不仅可以定位篡改位置,还可以对数据进行恢复[9-11]。

空间拓扑关系是空间关系研究的基础问题,也是空间推理、查询与分析的基础,并已被广泛应用到地理信息系统、导航和人工智能等方面。空间对象间的拓扑关系是指在平移、旋转、缩放等操作下其空间关系保持不变,即空间对象间的邻接、关联和包含关系不改变。

然而,现有的大部分矢量地理数据弱水印方法的研究都基于数据完整性,即任何对数据的修改(如几何操作、顶点修改和地物修改等)都将被视为篡改并被检测出来。但是,几何操作(如平移、旋转和缩放)并不改变地物之间的空间拓扑关系,即地物之间的邻接、关联和包含关系。只有当发生顶点篡改和地物篡改时,地物间的邻接、关联和包含关系才会发生改变,且被视为矢量地理数据的拓扑完整性被破坏。但是,几何操作可以通过其相应的逆操作得到原始数据,所以本文提出

到稿日期:2015-09-15 返修日期:2015-11-03 本文受国家自然科学基金项目(61262021)资助。
隋莉莉(1990-),女,硕士生,主要研究方向为空间数据安全,E-mail:lily289sui@sina.cn;汪传建(1977-),男,博士,副教授,CCF会员,主要研究方向为数据安全,E-mail:wcj_inf@shzu.edu.cn(通信作者)。

一种基于弱水印的地理数据拓扑完整性检验方法。通过地物之间空间相离距离生成水印信息，并修改其相离距离值，根据修改前后的相离距离的比值进行地物缩放，以达到嵌入水印的目的。而水印检测是通过判断地物间的空间拓扑关系是否发生改变，来验证矢量地理数据的拓扑完整性。

本文第 2 节描述了地理对象之间的空间拓扑关系以及计算拓扑关系的方法；第 3 节详细阐述了算法过程；第 4 节对算法性能进行了简单的讨论；第 5 节是实验结果；最后是总结与展望。

2 拓扑关系描述及计算

在 Egenhofer 提出的 9 交叉模型(9 Intersection Model)中，任意两个原子地理对象 A 和 B 之间的拓扑关系即为 A 的内部(A°)、边界(∂A)、外部(A^-)与 B 的内部(B°)、边界(∂B)、外部(B^-)的比较[2]。任意两个对象的 3 种位置可以组成 9 个基本拓扑关系的描述，并构成一个 3×3 的矩阵，即 9 交叉模型，如式(1)所示。

$$R(A,B)=\begin{pmatrix} A^\circ\cap B^\circ & A^\circ\cap\partial B & A^\circ\cap B^- \\ \partial A\cap B^\circ & \partial A\cap\partial B & \partial A\cap B^- \\ A^-\cap B^\circ & A^-\cap\partial B & A^-\cap B^- \end{pmatrix} \tag{1}$$

模型中的任意两个地理对象的拓扑关系具有几何不变性，即平移、旋转和缩放操作不会改变地理对象之间的拓扑关系。当以地理对象 A 为参考对象时，用于衡量 A 和 B 之间的相离关系的计算方法如式(2)所示。

$$EC(B)=\frac{area(B)}{area(B\oplus dist(\partial B,\partial A))} \tag{2}$$

其中，$area(B)$为 B 的面积，$dist(\partial B,\partial A)$表示 A 和 B 间的最短距离，$\oplus$为面积膨胀操作符，$area(B\oplus dist(\partial B,\partial A))$表示 B 膨胀 $dist(\partial B,\partial A)$之后的面积。相离距离值的计算示意图如图 1 所示。

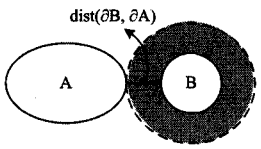

图 1 相离距离值计算示意图

3 算法描述

3.1 算法框架

基于弱水印矢量地理数据的拓扑完整性验证算法分为两部分：水印嵌入算法和水印检测算法。在水印嵌入算法中，首先是根据地理数据集 F 的空间拓扑关系生成水印串，然后将生成的水印串嵌入到地理数据集中。通过对地理数据的轻微修改，来达到水印嵌入的目的。水印嵌入流程如图 2 所示。

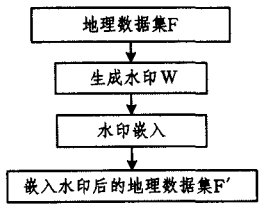

图 2 水印嵌入流程

水印检测算法用来验证地理数据集的拓扑完整性。首先，通过地理数据的空间拓扑关系生成水印串，同时提取原水印串，然后将生成的水印串与提取出的原水印串进行比较。若两者相同，则表明该地理数据集的拓扑关系没有被破坏；反之，则表明该地理数据集的拓扑关系已被破坏，其数据已被篡改。具体水印检测流程如图 3 所示。

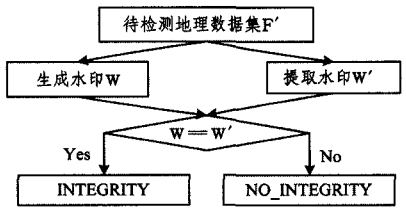

图 3 水印检测流程

3.2 水印嵌入

地理数据集中通常包含许多个地理对象，我们称之为地物，而每个地物是由许多个顶点组成的。假设地理数据集中的地物集合为 $F=\{F_1,F_2,\cdots,F_m\}$，m 表示地物的个数；每个地物中包含的顶点集合表示为 $F_i=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，$n$ 表示地物 F_i 中顶点的个数。

(1)计算地物间的最短距离

根据地物集中的两个相离地物计算出它们之间的最短距离。假设需要嵌入水印的地物为 B，地物 A 为地物 B 的参照地物，则两地物之间相距最近的顶点的距离即为地物 A 与地物 B 之间的最短距离，其计算方法如下：

$$dist(\partial B,\partial A)=\sqrt{(x_B-x_A)^2+(y_B-y_A)^2} \tag{3}$$

其中，点(x_A,y_A)与点(x_B,y_B)分别表示地物 A 与地物 B 之间相距最近的顶点。

(2)计算空间拓扑关系相离距离值

对需要嵌入水印的地物 B 与参照地物 A 进行空间相离距离值的计算，结果用 D 表示，其计算方法如下：

$$D=\frac{area(B)}{area(B\oplus dist(\partial B,\partial A))} \tag{4}$$

其中，$area(B)$为地物 B 的面积，$area(B\oplus dist(\partial B,\partial A))$表示 B 膨胀 $dist(\partial B,\partial A)$之后的面积，进而得到相离距离值集 $D=\{D_1,D_2,\cdots,D_k\}$，k 为水印串的长度。

(3)生成水印位

水印的生成是通过空间拓扑关系相离距离值 D_i 计算得到的。首先取 D_i 的整数位，并将其对 2 取模，如式(5)所示。根据得到的结果生成水印位 w_i，则水印串 $W=\{w_1,w_2,\cdots,w_k\}$。

$$w_i=Integer(D_i)\bmod 2 \tag{5}$$

(4)水印嵌入

根据水印位 w_i 对相离距离 D_i 的最低有效位进行修改。当水印位为 1 时，将 D_i 的最低有效位的值 $LSB(D_i)$修改至 5～9的范围内；当水印位为 0 时，将 D_i 的最低有效位的值 $LSB(D_i)$修改至 0～4 的范围内，如式(6)所示。地物 B 将按调整后的相离距离 D_i'与 D_i 的比值 s 进行缩放，从而实现水印的嵌入。水印嵌入算法的全过程如表 1 所列。

$$LSB(D_i)=\begin{cases} 5\sim 9, & w_i=1 \\ 0\sim 4, & w_i=0 \end{cases} \tag{6}$$

表 1　水印嵌入算法

```
Input: the original dataset F= {F1,F2,…,Fm}
Output: the watermarked dataset F'= { F1',F2',…,Fm'}
1. for each polygon Fi in F
2.   dist(Fi,Fi+1)
     //计算地物 Fi 与 Fi+1之间的最短距离
3.   if dist == 0 then i++
4.   Di←compute_measurement_distance(Fi,Fi+1)
     //计算地物 Fi 和 Fi+1的相离距离值
5.   if Di== 0 then i++
6.   wi←Integer(Di) mod 2//生成水印位
7.     Di'←modify(LSB(Di),wi)
       //根据 wi 的值修改 Di 的最低有效位
8.   s←Di'/Di//计算缩放因子
9.   scale(Fi,s)//缩放 Fi 以实现水印嵌入
10.   i++
11. endfor
```

3.3　水印检测

水印检测过程是用来判断地理数据集的拓扑完整性。首先计算地理数据集中相离的两个地物之间的最短距离，并计算两地物之间的空间相离距离值 D_i，由相离距离值 D_i 的整数部分生成水印位 w_i；然后根据相离距离值的最低有效位提取出原水印位 w_i'，并将原水印位 w_i' 与生成的水印位 w_i 进行比较。若相同，则判定该地理数据集的拓扑关系是完整的；否则，判定该地理数据集已被篡改。具体的检测算法如表 2 所列。

表 2　水印检测算法

```
Input: the suspicious dataset F'= { F1',F2',…,Fm'}
Output: INTEGRITY or NO_INTEGRITY
1. for each polygon Fi' in F'
2.   dist(Fi',Fi+1')
     //计算地物 Fi'与 Fi+1'之间的最短距离
3.   if dist == 0 then i++
4.   Di←compute_measurement_distance(Fi',Fi+1')
     //计算地物 Fi'和 Fi+1'的相离距离值
5.   if Di== 0 then i++
6.   wi←Integer(Di) mod 2 //生成水印位
7.   wi' ←quantization( LSB(Di) )
     //提取 Di 的最低有效位,并将其量化到 0 或 1
8.   if wi== wi' then right++
9.   i++
10. endfor
11. if right == k return INTEGRITY //k 为原水印串的长度
12. else return NO_INTEGRITY
```

3.4　拓扑完整性检测

根据生成的水印信息与原水印信息的匹配结果，来判定该地物的拓扑关系是否被篡改，从而判定该份矢量地理数据集的拓扑关系是否完整。

4　讨论

4.1　算法特点

在 Egenhofer 的 9 交叉模型中，任意两个地物之间都存在拓扑关系的几何不变性。当对地图进行平移、旋转和缩放操作时，并没有改变地物间的拓扑关系，而且平移、旋转和缩放操作可以通过其相应的逆操作得到原地理数据集，且不影响原地理数据集的正常使用。但是，如果对地物进行增加顶点、删除顶点、修改顶点甚至是增加地物、删除地物等操作，则会影响地物之间的拓扑关系，也就意味着破坏了地物之间拓扑关系的几何不变性。

相离距离值的计算是将地物的原面积与膨胀后的面积进行相比得到的。由相离距离值的整数部分生成水印信息，当地物受到几何修改(平移、旋转和缩放)时，相离距离值并不会受到影响，因为地物的面积没有发生改变。同时，地物之间的拓扑关系也没有发生变化，这就体现了水印在几何变换中的稳定性；当地物受到非几何修改时，相离距离值会受到影响，使得在水印检测过程中生成的水印信息无法与水印嵌入过程中生成的水印信息保持一致，即地理数据集的拓扑完整性已被破坏，从而体现了水印的脆弱性，并实现了地理数据集的拓扑完整性检测的目的。

4.2　算法的通用性

首先，本算法只适用于面类地理数据集。因为计算相离距离时，需要计算地物的面积，而点类地物和线类地物不存在面积值，也就无法得到相离距离值。因此，点类地物和线类地物既无法生成水印信息，也不能用缩放地物的方法嵌入水印信息。其次，本方法只适用于计算相离地物间的距离，因为地物的缩放需要一定的缓冲区间，如果地物之间的关系是相交或者相切，即两地物之间的最短距离值为 0，则无法对地物进行水印的嵌入，那么就需要更改需嵌入水印的地物，或者更换需嵌入水印地物的参考地物。

4.3　时间复杂度与空间复杂度

本文提出的方法在时间与空间复杂度上都是线性的。假设地理数据集中地物的数量为 m，则在水印嵌入和检测时，本算法的时间复杂度与空间复杂度均为 $O(m)$。

5　实验与结果

对矢量地理数据集进行平移、旋转和缩放等操作时，并不影响矢量地理数据集的拓扑完整性，即对矢量地理数据集进行几何变换不会造成数据集完整性的破坏。然而非几何操作(如化简、噪音等)会影响到地物之间的拓扑关系，进而矢量地理数据集的拓扑完整性会被破坏，所以为了验证算法对矢量地理数据集的拓扑完整性的检测效果，进行了以下一系列的相关实验。

实验用的矢量地理数据集是面类数据，地物数量 3515 个，顶点总数 119277 个，数据集如图 4 所示。实验分别对该数据集进行了化简、噪音、修改地物、增加地物和删减地物等篡改操作。当地理数据集被篡改时，检测出的水印位会与嵌入时的水印位产生无法完全匹配的现象。实验中，将不匹配的水印位的数量与原嵌入水印位的数量之比称为水印错误率。

图 4　地理数据集

5.1 几何修改

当对矢量地理数据集进行平移、旋转和缩放操作时，只需要对数据集进行相应的逆操作就可得到原始的数据集，而且此类操作并不影响地理数据集的使用。

5.2 化简

化简操作是将矢量地理数据集中的数据按照顶点总数的一定比例进行地物顶点的删减，其实验结果如图 5 所示。

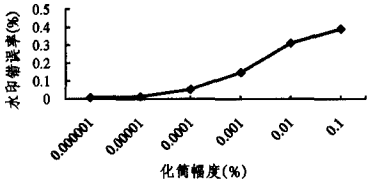

图 5 化简实验结果

从图 5 的实验结果中可以看出，随着化简幅度的逐渐增大，未匹配上的水印位也逐渐增多，说明该矢量地理数据集中有越来越多的地物面积被改变，而且地物间的拓扑关系遭到破坏的程度也越来越大。

5.3 噪音

噪音操作是对地理数据集中的数据按照顶点总数的一定比例进行顶点扰动，其实验结果如图 6 所示。

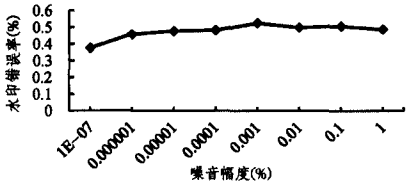

图 6 噪音实验结果

从图 6 的实验结果中可以看出，对地物顶点的轻微扰动，改变了地物的原始面积，使得检测出的水印无法与原水印匹配，从而表明地物之间的拓扑关系已遭到破坏。

5.4 修改地物

修改地物的操作是按照地物总数的一定比例随机选择地物，并将被选择的地物顶点进行随机扰动，其实验结果如图 7 所示。

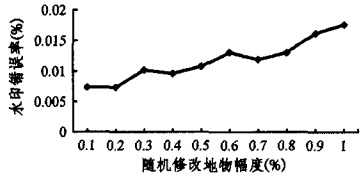

图 7 随机修改地物的实验结果

从图 7 的实验结果中可以看出，被选择的地物因被修改，其空间拓扑关系也随之发生了变化，使得拓扑完整性遭到破坏，而且随着被修改地物数量的增加，该地理数据集的拓扑完整性也逐渐降低。

5.5 增加地物

增加地物操作是按照地物总数的一定比例增加地物数量，地物被添加的位置是随机选择的，其实验结果如图 8 所示。

从图 8 的实验结果中可以看出，因为随机增加的地物影响了原地物之间的参照关系，所以在水印检测时，生成的水印无法与原水印完全匹配，表明该矢量地理数据集的拓扑完整性已经被破坏。

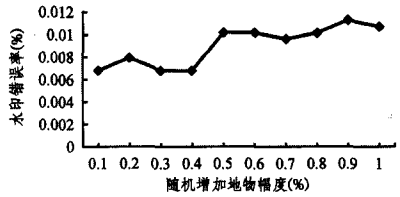

图 8 随机增加地物的实验结果

5.6 删减地物

删减地物操作是按照地物总数的一定比例随机选择被删减的地物，其实验结果如图 9 所示。

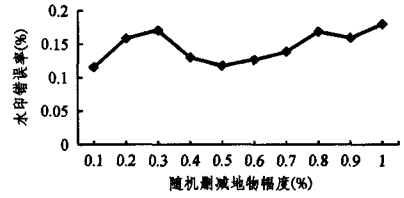

图 9 随机删减地物实验结果

从图 9 的实验结果中可以看出，随机删减地物，使得原地物之间的参照关系发生了改变，从而影响了空间相离距离值的计算，所以在水印检测时生成的水印无法与原水印完全匹配，从而表明该矢量地理数据集的拓扑完整性已被破坏。

结束语 本方法可以有效地对矢量地理数据的拓扑完整性进行检验，而且水印的嵌入对地理数据集的影响可以忽略不计。但是本方法只能适用于面类数据，今后将不断完善本文方法，使其可以适用于点类和线类的矢量地理数据集。

参 考 文 献

[1] 朱长青，王奇胜. 脆弱水印技术及其在矢量地理数据更新中的应用[J]. 测绘通报，2012(S1)：687-689

[2] Egenhofer M J，Herring J. Categorizing binary topological relations between regions，lines and points in geographic databases [R]. Santa Barbara CA National Center for Geographic Information and Analsysis Technical Report，1990：1-28

[3] Zheng Liang-bin，Jia Yu-lu，Wang Qun. Fragile digital watermark used to verify integrity of vector map[J]. Computer Engineering and Applications，2010，46(26)：99-101(in Chinese)
郑良斌，贾玉禄，王群. 用于矢量地图完整性验证的脆弱数字水印算法[J]. 计算机工程与应用，2010，46(26)：99-101

[4] Zheng Liang-bin，You Fu-cheng. A Fragile Digital Watermark Used to Verify the Integrity of Vector Map[C]// 1st International Conference on E-Business and Information System Security. 2009

[5] Niu Xia-mu，Shao Cheng-yong，Wang Xiao-tong. GIS Watermarking：Hiding Data in 2D Vector Maps[J]. Studies in Computational Intelligence，2007，58：123-155

[6] Wang Qi-sheng，Zhu Chang-qing. Fragile Watermarking Algorithm for Vector Geographic Data Exact Authentication[J]. Journal of Geomatics Science and Technology，2012，29(3)：218-221，225(in Chinese)
王奇胜，朱长青. 一种用于精确认证的矢量地理数据脆弱水印算法[J]. 测绘科学技术学报，2012，29(3)：218-221，225

[7] Chen Fan，He Hong-jie，Tai Heng-Ming，et al. Chaos-based self-

embedding fragile watermarking with flexible watermark payload[J]. Multimedia Tools and Applications, 2014, 721

[8] Wang Chuan-jian, Peng Yu-wei, Zhao Qing-zhan, et al. Fragile marking Scheme for Checking the Tamper of Geographical Data [J]. Mini-micro Systems, 2014, 35(12): 2628-2631(in Chinese)
汪传建,彭煜玮,赵庆展,等. 基于弱水印的地理数据篡改检验方法[J]. 小型微型计算机系统,2014,35(12):2628-2632

[9] Neyman S N, Sitohang B, Sutisna S. Reversible Fragile Watermarking based on Difference Expansion Using Manhattan Distances for 2D Vector Map[C]// The 4th International Conference on Electrical Engineering and Informatics(ICEEI 2013). 2013:614-620

[10] Wang Na-na, Men Chao-guang. Reversible fragile watermarking for 2-D vector map authentication with localization[J]. Computer-Aided Design. 2012, 44(4): 320-330

[11] Wang Na-na, Men Chao-guang. Reversible fragile watermarking for locating tampered blocks in 2D vector maps[J]. Multimedia Tools and Applications, 2013, 67(3): 709-739

(上接第 158 页)

漏洞,DSTA 只需生成该漏洞所在路径的某个测试用例,所以 DSTA 所需要的测试用例也相对更少。

表 4 可执行程序漏洞检测

漏洞编号	软件版本	DTA	DSTA	
			漏洞类型	漏洞位置
CNVD-2008-2394	WordNet 2.1	No Vulnerability	Buffer Overflow	searchwn()
CNVD-2009-5871	Xpdf 3.04	No Vulnerability	Buffer Overflow	FoFiType1::parse()
CNVD-2010-1209	Scite 2.26	No Vulnerability	Buffer Overflow	ReadLine()
CNVD-2014-01613	Freetype 2.5	No Vulnerability	Buffer Overflow	cf2_hintmap_build()

4.3 讨论

从以上实验结果可知,该系统不仅获得了正确的信息,同时可以检测出传统 DTA 无法检测出的风险,具有更大的实用性和高效性。但是该系统还存在以下不足:一方面,该系统只跟踪了数据流,而没有跟踪控制流,导致无法检测出由控制流产生的风险;另一方面,该系统建立在插装平台之上,导致检测程序时速度较慢。这些缺陷将在后续的工作中进行研究并加以解决。

结束语 本文介绍了动态符号化污点分析的思想,同时基于 Pin 插装平台实现了一个原型系统。DSTA 是对 DTA 的一种符号化处理及改进,并且取得了更好的效果:一方面它不仅可以直接检测出程序中实际发生的漏洞,同时也可以检测出在程序运行结束前该路径下潜在的风险情况;另一方面,该系统获得的风险变量集可以用于生成测试用例来验证该风险是否真的会触发漏洞,通过针对性地产生测试用例来提高测试效率,降低冗余。

未来的工作包括:完善该系统下的不同类型漏洞的风险检测;跟踪数据流的同时跟踪控制流,并且在跟踪控制流的同时获得该路径分支约束条件,以提高该系统漏洞检测的路径覆盖率;根据风险变量集的报告自动生成测试用例,实现系统的自动化;提高程序检测的速度。

参 考 文 献

[1] Peach [EB/OL]. http://peachfuzzer.com/. 2009 June

[2] SPIKE [OL]. http://www.immunitysec.com/resources-free-software.shtml

[3] Luk C K, Cohn R, Muth R, et al. Pin: building customized program analysis tools with dynamic instrumentation[J]. ACM Sigplan Notices, 2005, 40(6): 190-200

[4] Nethercote N, Seward J. Valgrind: a framework for heavyweight dynamic binary instrumentation [J]. ACM Sigplan Notices, ACM, 2007, 42(6): 89-100

[5] Newsome D S J. Dynamic Taint Analysis: Automatic Detection, Analysis, and Signature Generation of Exploit Attacks on Commodity Software[C]// Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS). 2005

[6] Qin F, Wang C, Li Z, et al. Lift: A low-overhead practical information flow tracking system for detecting security attacks[C]// 39th Annual IEEE/ACM International Symposium on Microarchitecture, 2006(MICRO-39). IEEE, 2006: 135-148

[7] Clause J, Li W, Orso A. Dytan: a generic dynamic taint analysis framework[C]// Proceedings of the 2007 international symposium on Software testing and analysis. ACM, 2007: 196-206

[8] Bekrar S, Bekrar C, Groz R, et al. A taint based approach for smart fuzzing[C]// 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST). IEEE, 2012: 818-825

[9] King J C. Symbolic execution and program testing[J]. Communications of the ACM, 1976, 19(7): 385-394

[10] Niu Wei-na, Ding Xue-feng, Liu Zhi, et al. Vulnerability Finding Using Symbolic Execution on binary program[J]. Computer Science, 2013, 40(10): 119-121, 138(in Chinese)
牛伟纳,丁雪峰,刘智,等. 基于符号执行的二进制代码漏洞发现[J]. 计算机科学,2013,40(10):119-121,138

[11] Kang M G, McCamant S, Poosankam P, et al. DTA++: Dynamic Taint Analysis with Targeted Control-Flow Propagation [OL]// www.cs.berkeley.edu/~dawnsong/papers/2011%20dat++-ndss11.pdf

[12] Wang T, Wei T, Gu G, et al. TaintScope: A checksum-aware directed fuzzing tool for automatic software vulnerability detection [C]// 2010 IEEE Symposium on Security and Privacy (SP). IEEE, 2010: 497-512

[13] Wang Z, Tang Z, Zhou K, et al. DsVD: An Effective Low-Overhead Dynamic Software Vulnerability Discoverer [C]// 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS). IEEE, 2011: 372-377

[14] Pin[OL]. https://software.intel.com/en-us/articles/pintool

[15] Kemerlis V P, Portokalidis G, Jee K, et al. libdft: Practical dynamic data flow tracking for commodity systems[J]. ACM SIGPLAN Notices, ACM, 2012, 47(7): 121-132