信息战环境下基于数据依赖的文件可靠恢复*)

陈伟鹤 殷新春 谢 立

(南京大学计算机科学与技术系 计算机软件新技术国家重点实验室 南京210093)

摘 要 虽然现有的安全操作系统能够防止非授权用户的访问,但是它们不能阻止授权用户的恶意攻击行为。在信息战环境下,恶意授权用户(malicious authorized user)发起的数据篡改攻击是一种新的严重安全威胁。它通过被恶意修改的数据误导被攻击的组织做出错误的决策。针对恶意系统授权用户造成的文件数据篡改破坏问题,本文提出了一种基于数据依赖的文件数据可靠恢复算法。在发现系统授权用户的恶意攻击行为后,它能够通过对恶意用户所攻击的文件数据和非恶意用户所访问文件数据间存在的数据依赖关系的分析,自动发现被破坏的数据并对其进行自动修复。它的优点在于对受破坏的文件数据恢复时,能够保留未受恶意攻击影响的工作,从而提高系统的可用性,增强抗恶意攻击的能力。

关键词 信息战,安全操作系统,可靠恢复,文件,恶意授权用户,内部威胁

Data-Dependency Based Trusted Recovery of File in Information Warfare

CHEN Wei-He YIN Xin-Chun XIE Li

(Department of Computer Science and Technology Nanjing University Nanjing 210093)

(State Key Lab. for Novelsoftware Technology Nanjing University Nanjing 210093)

Abstract Although existing secure operating systems can prevent unauthorized users' access requirements, it is out of its' reach to stop authorized users' malicious attacks. In information warfare context, malicious modification of data values by system authorized users is a more dangerous threat. It misleads the attacked organization into making wrong decisions by using the malicious modified data values. In order to solve the problem of malicious modification of file data caused by malicious authorized users, a data-dependency based file data damage assessment and trusted recovery algorithm is presented in this paper. After attacking behaviors of malicious authorized users were found, it can do file data damage assessment and trusted recovery automatically by analyzing the data dependency existing among the data values written by malicious attackers and the data values read/written by benign users.

Keywords Information warfare, Secure operating system, Trusted recovery, File, Malicious authorized user, Insider threat

1 引言

信息战[1,2]通过基于信息技术手段的攻击,使被攻击组织的信息系统不能正常工作以达到攻击者的目的。攻击的目标可能是信息系统本身或者是其中的数据。在信息战环境下,一个对信息系统较具危害性的破坏是攻击者通过没有被及时发现的入侵行为,伪装成合法的授权用户,通过对某些数据的恶意修改,在系统中制造一些不正确的数据,利用它们误导被攻击的组织作出有利于攻击者的决策[2]。

在传统的信息安全研究[3]中,认为没有获得授权的用户的行为都是违背系统安全策略的,应该禁止;而授权用户的符合授权的行为都是不会造成系统破坏的行为。但是这种逻辑假定实际上是存在安全隐患的,因为可能面临着两类系统授权用户的恶意行为。首先,计算机网络消除了信息系统内部用户和外部用户之间原来存在的界线。因此许多基于网络的攻击,通过密码嗅探(password sniffing)和会话劫持(session hijacking)等手段可以使一个攻击者伪装成获得授权的合法系统用户。其次,出于经济、宗教信仰、政治等原因,系统授权用户中也完全可能出现一些变节者。

由信息系统恶意授权用户构成的内部安全威胁是传统的访问控制等基于预防的安全机制所无法解决的。恶意的系统授权用户通过修改数据信息,误导数据的使用者,达到恶意攻击的目的。因此,研究如何防止系统授权用户可能对数据造成的破坏具有十分重要的现实意义,它提高了信息系统的生存能力、抗攻击能力,也是信息战和信息保障所要求的。

文件作为组织数据的一种重要方式,它的安全和抗恶意攻击能力在信息战环境下具有特殊的意义。虽然可以采用数字签名等方式防止文件被非法篡改,但是对于系统授权用户对数据的恶意修改,数字签名就无能为力了。针对恶意系统授权用户造成的文件数据非法篡改问题。在本文中,我们提出了一种基于数据依赖的文件可靠恢复算法。它能够自动确定受到恶意攻击破坏的数据,并且对被破坏的数据进行自动恢复。而且在对受到破坏的数据进行恢复的同时,能够保留未受恶意篡改攻击影响的工作,从而提高了系统的可用性和抗恶意攻击能力。

2 算法

2.1 问题的提出和基本解决思路

在信息战环境下,实现文件系统的抗恶意篡改攻击是一件较有意义的事。首先,计算机在信息系统构成和信息战攻防

*)本文得到国家"863"高技术(NO:2001AA144010)经费资助。陈伟鹤 博士研究生,主要研究方向为信息安全和分布式系统。殷新春 博士研究生,主要研究方向为信息安全和分布式系统。谢 立 教授,博士生导师,主要研究领域为信息安全和分布与并行计算机系统。

中都是一个重要组成部分。操作系统作为基础软件平台，它的安全性和抗恶意攻击能力的强弱对整个信息系统的安全性和可生存性(survivability)有重要影响。虽然可以采用传统的备份恢复的方法，但是它在将文件系统数据恢复到被攻击前的最近一个备份点以消除恶意篡改攻击所造成影响的同时也使实际未受攻击影响的工作被撤消掉了。这对系统可用性造成了严重影响，降低了系统抗恶意攻击的能力。更值得注意的是，这种由于系统恢复而撤消未受攻击影响的工作的情形可能正是恶意攻击者所期望的[1,2]。因此，在信息战环境中，当计算机系统受到恶意攻击时，只要文件数据并没有被破坏到难以恢复的程度，操作系统就应该有相应的机制对受到破坏的文件数据进行定位并能够可靠地自动恢复到遭受破坏之前的状态，且保留未受恶意攻击影响的工作。

其次，由于入侵检测[4]机制能够发现异常行为，在信息战环境下，为了取得理想的恶意攻击效果，恶意授权用户的攻击必须以尽量隐蔽的方式进行，缩小攻击范围，只对重要数据攻击，降低被发现的风险[1,2]。由于信息战环境下的恶意授权用户攻击通常具有这个特点，因此，在信息战中，操作系统中的文件通常并不会在恶意攻击中受到全部破坏，往往只是若干个文件，或某个文件中的若干个部分受到攻击。另一方面，许多操作系统中的应用并不是涉及到所有的操作系统文件，而只是访问若干个特定的文件。甚至，可能只是访问某个文件的某一个部分所记录的数据。因此，实现基于数据依赖的文件数据可靠恢复是可能的。所以，无论是出于抗恶意攻击的目的还是从提高信息系统的可用性考虑，都有必要实现文件数据的自动可靠恢复。

本文提出的文件数据可靠恢复方法的基本思想是：对操作系统进行扩充，添加若干个数据结构，它们的作用在于记录用户对文件的操作，为文件数据恢复提供必要的信息。操作系统在处理用户访问时，记录用户访问的文件名字、时间、操作及操作所涉及到的数据的原有值和访问后的新数据值。当操作系统通过攻击检测技术，发现某个用户从某一时刻开始成为恶意用户时，从此时起，他所访问的那些文件就认为是受到了恶意破坏(如果此用户对某些文件只是作了读数据(read)操作，那么可以认为这些文件并没有受到破坏)。而且随着其它无辜用户直接/间接访问被破坏的数据项，恶意攻击造成的数据破坏范围可能进一步扩大。通过分析用户所访问的文件数据之间存在的数据依赖关系，能够实现被破坏数据项的自动确定和恢复，而未受攻击影响的数据和相应的工作则不受数据恢复工作的影响。在文件数据恢复时，没有受到恶意攻击破坏的其它文件可以继续供授权用户所使用。而且，一旦确定了文件中的哪些数据受到破坏，在操作系统对被破坏的文件数据进行修复的同时，可以把未受破坏的数据提供给后续用户访问。这就进一步提高了系统的可用性。

2.2 模型

文件[5]是由多个性质相同的记录组成的集合。在本文中将数据粒度细化到记录级。文件系统可以看作是文件、目录以及定义在它们上面的操作所构成的集合。设有一个文件系统，它包含一系列文件，记为：$f_1, f_2, \cdots, f_n$。每个文件都定义了一组操作，它是 read、write、create、delete、append 等构成的一个集合。设系统有 m 个用户，分别记为：$U_1, U_2, \cdots, U_m$。将他们各自所对应的操作序列记为：

$O_{i1}, O_{i2}, \cdots, O_{in}, \cdots,$

其中，$1 \leqslant i \leqslant m$

对任意一个操作 O，它都包含了以下信息：实施该操作的用户、操作的类型(即 read、write、create、delete 等)、操作所涉及的文件、操作所要访问的数据项在文件中的位置以及数据项在操作前后的值。

为了后面描述方便起见，在本文中也用$[U_i, x, v_1, v_2]$表示用户 U_i 的写操作，$[U_i, x, v]$表示用户 U_i 的读操作。其中，U_i 表示第 i 个用户；x 称为数据项，它包含操作所涉及的文件以及操作所要访问的字段在文件中的位置等等信息；v_1表示数据项 x 的前像(before image)，即数据项 x 的原有数据值；v_2表示数据项 x 的后像(after image)，即数据项 x 的新值。在不强调数据项值的情况下，本文中也采用 $W_{ui}[a]$、$R_{ui}[a]$分别表示用户 U_i 写、读数据项 a。写操作有时也称为更新操作。

2.3 相关概念定义

类似于文[7,9]，我们定义用户操作执行历史和可恢复用户操作执行序列如下：

定义1(用户操作执行历史) 它按照用户操作执行的顺序，记录了用户对操作系统中文件的所有操作，它不能够被用户所修改，因此它所记录的数据是可靠的。

定义2(可恢复用户操作执行序列) 如果用户操作执行历史中，存在着一段用户操作执行序列，其中记录的对文件的操作都是对文件数据读、写(更新)操作，而不涉及删除、追加数据等操作，那么称这段用户操作执行序列是可恢复用户操作执行序列。

由于难以获取用户文件操作的精确语义信息，在本文中我们假定任何一个用户的写操作受到该用户在本次文件操作会话(session)中在此写操作之前得到执行的读操作的影响，即定义同一用户读写操作间的依赖关系如下：

定义3(读操作和写操作之间的依赖关系) 在一次文件操作会话中，如果 $R_{ui}[x]$在 $W_{ui}[y]$之前得到执行，称用户 U_i 的写操作 $W_{ui}[y]$依赖于读操作 $R_{ui}[x]$。记为：$R_{ui}[x] \rightarrow W_{ui}[y]$，其中读/写操作的数据项可以是同一个。

定义4(数据值之间的依赖关系) 设有数据项 v_1、v_2，如果产生数据项 v_2的写操作依赖于获取数据项 v_1数据值的读操作，那么称数据项 v_2的数据值依赖于数据项 v_1的数据值。记为：$v_1 \rightarrow v_2$。

定义5(修复赋值写) 如果非恶意用户 U_i 对某个数据项 x 值的更新不依赖于任何受到恶意攻击破坏的数据项的值，则称此写操作 $W_{ui}[x]$为修复赋值写。

定义6(数据项的正确值) 如果在没有出现恶意攻击情况下，数据项 x 的值为 v，那么，称数据值 v 为数据项 x 的正确值。

为了判断读/写操作之间的相互依赖关系，在用户操作执行历史中保存了用户的所有读/写操作。尽管读操作所获得的数据项的值可以从用户操作执行历史中该操作之前的对同一数据项的最后一次更新的后像(after image)获得，但为了优化系统的性能，提高处理速度，把读操作和所读数据项的值都保存在用户操作执行历史中。

例1 设有可恢复用户操作执行序列 H_1

$H_1 = R_{u1}[a]R_{u1}[b]W_{u1}[c]R_{u2}[a]W_{u2}[b]W_{u2}[d]R_{u3}[d]R_{u3}[a]R_{u3}[c]W_{u3}[d]R_{u4}[b]W_{u5}[a]W_{u5}[b]R_{u6}[b]W_{u6}[b]R_{u6}[c]W_{u6}[c]R_{u6}[d]W_{u6}[d]R_{u6}[a]W_{u6}[a]$

我们采用 $p = U_i(q_1, q_2, \cdots, q_n)$表示用户 U_i 所更新的数据项 p 的值依赖于用户 U_i 所读取的数据项 $q_1, q_2, \cdots, q_n$ 的值。

根据前面定义的数据依赖和操作依赖，对 H_1我们可以得到数据值间存在如下的依赖关系：$c = U_1(a,b)$，$b = U_2(a)$，$d = U_2(a)$，$d = U_3(a,c,d)$，$a = U_5(\ )$，$b = U_5(\ )$，$a = U_6(b)$，$c = U_6(b,c)$，$d = U_6(b,c,d)$，$a = U_6(a,b,c,d)$。

设在系统运行中，通过入侵检测发现用户 U_2 是恶意授权用户，因此数据项 b 和 d 就认为受到了破坏。接下来，当用户 U_3 读取了数据项 d 的值后再给其赋新值时，d 还是被破坏的数据项。到此时为止，数据项 b 和 d 都被恶意破坏。当用户 U_5 对数据项 b 的值更新以后，因为 U_5 是善意用户，而且他在更新 b 之前没有读取任何被破坏的数据项 b 和 d 的值，因此，数据项 b 的值被修复为正确的数据值。当用户 U_6 更新数据项 b 和 c 的值时，没有读取数据项 d 的值，因此 b 和 c 的值都是正确的数据值。而当 U_6 更新数据项 d 的值时，由于之前读取了 d 被恶意破坏的值，所以数据项 d 的值依然是被破坏的。而且，由于用户 U_6 对数据项 a 的数据值的更新依赖于对数据项 d 的读操作，因此数据项 a 也受到恶意攻击的破坏。

2.4 基于数据依赖图的受恶意攻击破坏数据的确定

数据依赖图(Data-Dependency Graph)能够直观反映文件数据被破坏和修复的情况。数据依赖图定义为：

$DG(H)=(V,E)$

数据依赖图是有向图，由节点和有向边组成。其中，H 表示可恢复用户操作执行序列，V 表示节点，E 表示有向边。在数据依赖图中，每个节点代表一个数据项在某一特定时间点上的值，它包含了数据项的名字，数据值更新的时间，以及数据是否受到恶意攻击的破坏。在数据依赖图中存在两类节点：圆圈和正方形。其中，圆圈节点表示该数据项的值没有受到恶意攻击，而正方形节点则表示数据项的值受到了恶意攻击的破坏。数据依赖图中的有向边表示用户对数据项值的更新操作，描述了数据项值之间的数据依赖关系。

在构造数据依赖图时，对任意一个数据项 x 按照其被更新的时间顺序垂直向下添加新节点描述数据项 x 在此时刻数据值的变化，在这两个节点添加垂直向下的有向边，边标记为对数据项进行更新的用户 ID。图刚开始构造的时候都是圆圈节点，每个节点表示一个数据项。当发现一个恶意攻击后，对受到恶意攻击影响的每一个数据项添加一个正方形节点，同时在存在数据值依赖的每对节点间加一条带标记的有向边，标记为实现此次更新的用户 ID，有向边起始于读操作所访问的数据项，终止于写操作所访问的数据项。当用户对一个受到恶意攻击破坏的数据项进行修复赋值写后，在图中为此数据项添加一个圆圈节点，并相应的增加一条有向边。

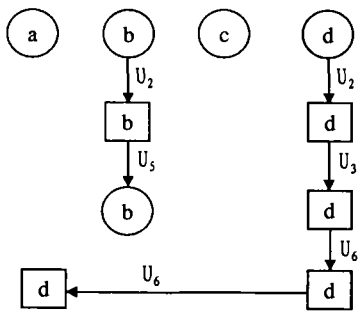

图1 例1的基于数据依赖图的受恶意攻击破坏数据的确定

从图1中可以看到，恶意攻击者 U_2 破坏了数据项 b 和 d 的值，但数据项 b 的被破坏的数据值随后就被用户 U_5 的修复赋值写修复了。对用户 U_6，由于数据项 a 的新值依赖于受到恶意破坏的数据项 d 的值，因此他对数据项 a 的数据值更新扩大了受破坏数据的范围。

2.5 文件恢复算法

correct_value_list：是记录列表，每个记录有两个字段：数据项字段和数据值字段。

read_item_list_U_i：是记录列表，每个记录有两个字段：数据项字段和数据值字段。记录了用户 U_i 所读取的数据项。

damage_item_list：记录了文件中被破坏的数据项。对其中的每个数据项，在 correct_value_list 中计算和保留了它们的正确值，这些值是如果恶意攻击没有发生情况下数据项本来应该拥有的值。恶意攻击造成的数据破坏的特点是随着时间的变化和非恶意用户对被破坏数据项值直接/间接读操作，引起被破坏的数据范围的扩大。因此，在 damage_item_list 中按时间顺序记录文件中被破坏的数据项。

Data-Dependency based File Damage Assessment and Trusted Recovery Algorithm

Input：攻击检测系统发现的恶意用户行为 O，用户操作执行历史 H
Output：修复了被恶意破坏的数据后的文件

```
Pass 1: 1 set damage_item_list={};//初始化数据结构
        2 set correct_value_list={};
        3 scan the user operation execution history from the first
          write operation record of the first malicious user until the
          end, for every operation O recorded in the history
        3.1 if O is a write operation [Ui,x,v1,v2] of a malicious
            attacker Ui
            if x∉damage_item_list
               { insert x into damage_item_list;
                 insert the record (x, v1) into correct_value_
                 list;}//v1是x的前像
        3.2 if O is a read or write operation of any other benign user
            Uj
            {3.2.1 if it is entering a new file operating session
                     set read_item_list_Uj={};
             3.2.2 if O is a read operation [Uj,x,v] of Uj
                   { insert record (x,v)into read_item_list_Uj if
                   x is not in it,otherwise update v;
                   if x∈damage_item_list
                     update value v of record (x,v)in read_item_
                     list_Uj by value of x in correct_value_list;}
             3.2.3 if O is a write operation [Uj,x,v1,v2] of Uj
                   {3.2.3.1 if the set of data item in read_item_
                              list_Uj∩damage_item_list≠∅
                            {recalculate new value v2 of x, by
                            using values in read_item_list_Uj;
                            3.2.3.1.1 if x∈damage_item_list
                                      update value of x in
                                      correct_value_list by new
                                      value v2;
                            3.2.3.1.2 else
                                      {insert record (x,v2)into
                                       correct_value_list;//x
                                       受到恶意攻击影响
                                      insert x into damage_ite-
                                      m_list;}
                            }
                    3.2.3.2 else{if x∈damage_item_list //对x
                            进行修复赋值写,恢复了x的正确数
                            据
                            {remove x from damage_item_list;
                            remove the record of x from the
                            correct_value_list;}
                            }
                   }
            }
Pass 2: for each item in damage_item_list
        {   all data items in the damage_item_list are blocked from
        being read by any active benign user, but a repair write on
        them by any benign user must be allowed;
            generated an isolated version of damaged files based on
        the damage_item_list, and redirect malicious authorized
        users' operation to it to mislead the attackers;
            update its value in the file by the value in the correct_
        value_list;
        }
```

在算法步3.2.3.1，为了重新计算 x 的值，需要知道用来计算 x 后像的逻辑操作。因此，在用户操作执行历史中保存该逻辑操作。一旦确定了文件中受到恶意破坏的数据，操作系统就阻止用户对这些数据的读取。这样就能够防止破坏的进一步扩散。但是，用户对这些受到破坏的脏数据的修复赋值写应该是允许的，因为它修复了受到破坏的数据项，而且反映了系统的最新状态。为了提高系统的可用性，及时提供数据访问服务，每个修复好的数据项都立即向用户提供访问服务。其次，根据被破坏的数据，产生一个隔离的虚拟系统，它包括被破坏

的文件，并且把已发现的恶意授权用户的操作都转换为对该虚拟系统的操作，达到欺骗、误导攻击者，并进行攻击取证的目的，限于篇幅有关攻击欺骗系统的具体描述将另文介绍。

算法的正确性证明比较明显，限于篇幅，此处不再赘述。

2.6 算法实现中的几个问题

(1)算法的前提条件。本算法基于与文[7,9]相同的前提条件，即对恶意授权用户的发现总是可以通过异常检测和其它一些攻击检测手段有效实现的。

(2)算法的实用性。显然算法的实用性与攻击检测的效率有很大关系。因此，算法的可行性可以从攻击检测技术的原理来说明。由于恶意攻击从实施到被发现的时间间隔是和受恶意攻击直接/间接影响的数据的数量相关的：受攻击的数据越多，被破坏的迹象越明显，就越有可能被发现，发现攻击的延迟就越短。因此，如果恶意攻击被发现的时间间隔较长，那么受破坏的数据通常也不会很多，或者说对现实世界的影响也不明显，必需恢复的数据量也不会很大。另外，随着技术进步，攻击检测的效率也在不断提高。还有一点值得注意，即不管是否采用我们所提出的文件数据恢复算法，消除已发生的恶意篡改攻击对现实世界的影响都是无法避免的。

(3)加入本算法后导致的系统负载增加问题。在最坏情况下(可恢复用户操作执行序列中的每个记录的内容都是一个新的恶意攻击，且该可恢复用户操作执行序列中有 N 个记录)，算法的时间复杂性为 $O(n^2)$，空间复杂性为 $O(n)$。本算法的主要弱点是需要操作系统提供大量存储空间来记录用户的所有文件操作，并保证所保存的操作记录的可靠。我们知道，现有的安全操作系统为了实施安全审计，同样需要保存用户的许多操作，并保证这些数据的可靠性。因此，就保证数据的可靠性而言，该算法并没有给安全操作系统引入新的问题，用于保证审计数据可靠性的方法同样适用。至于为了记录用户的所有文件操作而导致的对存储空间的大量耗费问题，可以通过对需要提供数据可靠恢复保护的文件数量加以限制的方法来大幅度降低对空间的消耗，即只保护少数重要的文件。当然，即使采取了这些措施也还是要占用较多存储空间，但这是实施信息保障，提高系统抗恶意攻击能力所必须付出的代价，对许多强调系统抗恶意攻击能力和必须提供连续服务的应用领域来说，这些付出还是值得的。

(4)算法的具体实现。我们目前在 Windows98操作系统上实现了一个原型模拟系统。它分为文件监控模块，数据破坏评估和自动恢复模块两大部分。文件监控模块基于 VxD 技术，它在动态加载后的初始化时利用 VxD 服务通过安装一个文件过滤器将其插入到文件系统的请求处理调用链中，从而实现对文件操作的监控。数据破坏评估和自动恢复模块根据文件监控模块提供的数据，在发现恶意攻击后进行数据依赖分析和恢复。我们做了三个实验。实验一对于同样的恶意攻击序列，选取不同的恶意攻击发现延迟时间，考察其对系统恢复的影响。实验表明随着恶意攻击发现延迟时间的缩短，由于本算法能够阻止对已发现的被攻击数据的读取以及恶意用户的后续操作，因此系统恢复所需时间也相应缩短。实验显示恶意攻击的频度、被攻击数据对象的集中还是分散、应用对文件访问的特点等都影响系统恢复的速度。随着攻击发生频率的提高，由于数据污染的蔓延，需要较长的系统恢复时间。这也表明随着攻击检测技术效率的不断提高，本算法的实用性也将不断改善。实验二固定恶意攻击被发现的时间延迟，随着需要保护的文件个数的变化，研究系统负载的变化。实验三则固定受保护的文件数量，随着发现恶意攻击的延迟时间间隔长度增加，研究对系统负载的影响。模拟实验表明通过限制需要提供可靠恢复保护的文件数量(在实际的安全应用中，对许多安全机制而言，都必须考虑性能和安全性的折中。因此，我们也不妨假定在实际应用中通常只需要保护极少数重要文件)，提高恶意攻击发现的效率，算法是能够取得较为满意效果的。攻击检测的效率是一个关系到算法实用性的重要因素，随着恶意攻击检测技术的进步，算法的实用性必将得到加强。

3 相关工作及比较

入侵检测技术在一定程度上能够起到克服访问控制等系统保护机制对恶意的授权用户无能为力的局限。但由于入侵检测存在的时间延迟，当一个入侵被确认时，破坏往往已经造成了。而且，入侵检测系统的入侵事件发现率是和系统的误报率成正比的。在许多计算系统中，如果要求入侵检测系统的准确率高就可能导致较高的入侵漏报率，并引起较长的时间延迟。由于许多入侵没有能够检测出来或者是有较长的时间延迟，就会导致系统受到一定程度的破坏。

在信息系统的抗恶意攻击研究中，受美国国防部 DARPA 的资助，近来有一些学者面向数据库管理系统和文件系统领域，开展了对恶意授权用户构成的安全威胁问题的探索，并发表了各自的研究结果[6~12]。这方面的研究属于信息系统的入侵容忍(intrusion tolerance，也称为弹性)技术，它是信息安全技术研究的前沿。入侵容忍技术假设攻击与正常数据是不能明确区分的，攻击的发生不可避免，研究如何在有攻击的情况下，使系统仍能为预期的合法用户提供可靠有效的服务。

针对数据库中存在的由恶意数据库用户导致的恶意攻击事务问题，Peng Liu[7,8]提出了基于事务的数据库入侵容忍技术。一旦通过攻击检测发现了恶意攻击事务，它们能够把恶意事务以及其它受到直接和间接影响的事务对数据库状态的改变撤消掉，使数据库就像没有发生过恶意攻击一样。它可以使攻击发生后未受恶意事务影响的无辜事务的工作结果保留下来。通过保留尽可能多的工作，提高系统的可用性。而 Branjendra Panda[9]的方法不是把受恶意攻击事务直接/间接影响的那些事务的所有操作都撤消掉(undo)，然后重新执行这些受影响的事务；而是只把这些受影响的事务中的受影响的操作撤消掉并重做。

另外，美国 GMU 的 Sushil Jajodia 等提出了在数据库环境中对可疑恶意用户进行隔离(user isolation)的机制[11,12]。它的主要思想是在应用层隔离可疑用户操作，而不是立即终止该用户提交的数据库事务操作，如果数据库管理系统在后续的操作中发现该数据库用户不是恶意攻击者时，数据库管理系统就能够以较少的资源消耗，达到保留该用户尽可能多的事务操作的目的。该方法把数据库分成主版本和嫌疑数据库版本。当数据库系统发现某个用户具有攻击嫌疑时，它就透明地把用户和主数据库版本隔离开来，防止可能造成的系统破坏的进一步蔓延。同时生成一个嫌疑数据库版本，并把该嫌疑用户的所有后续操作转换为对此嫌疑数据库版本的操作。当确定了该嫌疑用户不是恶意攻击者时，数据库管理系统将对应于该用户的嫌疑数据库版本和数据库主版本进行合并。从而达到既防止恶意攻击可能造成的数据破坏，又尽可能保留了用户正常事务操作的目的。

美国 UMBC 的 Peng Liu 提出了文件系统的侵入限制(intrusion confinement)[6]，他的基本想法是在一个可疑入侵行为被确认之前，采取预防措施，限制该行为可能对系统造成的破坏程度。他给出了一个文件系统隔离协议和对同一个文件的不同版本进行合并的文件系统合并协议，实现粒度为文

件级的隔离和数据更新的合并。每个没有被删除的文件都有一个主版本。系统中存在着同一个文件的多个版本，每个可疑用户有一个对应版本的文件。当一个可疑用户访问一个文件时，系统根据文件隔离协议产生相应的嫌疑版本的文件供其访问。当某个可疑用户最终被认定是非恶意用户时，就根据文件系统合并协议将文件的主版本和可疑版本进行合并，反之，则将嫌疑版本的文件抛弃。

但是，在信息战环境下当发生恶意授权用户攻击时如何实现文件数据的抗恶意攻击和自动恢复，并保留未受恶意攻击影响的工作，则是一个没有涉及的领域。

因此，我们提出了这种基于数据依赖的文件可靠恢复算法。在发现了恶意授权用户后，它能够自动确定受到恶意攻击破坏的数据，并且能对被破坏的数据进行自动恢复，而且在对受到破坏的系统进行恢复的时候能够保留未受恶意攻击影响的工作。同时能够通过建立一个虚拟的遭到破坏的文件系统环境，达到欺骗、误导恶意攻击者，并实现信息战反击的目的。

结论 随着对信息战环境下恶意授权用户构成的安全威胁严重程度的认识，为了弥补访问控制等安全预防机制和入侵检测技术的不足，信息系统的抗恶意授权用户攻击和攻击后破坏范围自动确认及自动恢复能力的研究已经成为新的研究热点。在信息战环境下如何实现文件数据级的抗恶意授权用户攻击，是一个没有得到应有重视的领域。针对系统恶意授权用户构成的文件数据非法篡改威胁。在本文中，提出了一种基于数据依赖的文件数据可靠恢复算法。它能够自动确定受到恶意攻击破坏的数据，对遭到破坏的数据进行自动恢复，并能够保留未受恶意攻击影响的工作。并就算法实现时的几个关键问题做了探讨。我们下一步的主要工作是研究发现恶意攻击的高效方法，提高算法的效率，增强算法实用性。

参 考 文 献

1 Waltz E. Information Warfare: Principles and Operations. Boston London, USA. Artech House, 1998
2 Jajodia S, Ammann P, McCollum C D. Surviving information warfare attacks. IEEE Computer, 1999, 32(4): 57～63
3 Bell D E, LaPadula L J. Secure Computer System: Unified Exposition and Multics Interpretation: [Technical Report MTR-2997]. MITRE, Bedford Masschusetts. , HQ Electron. Syst. Dov. , Hanscom AFB, MA, TEch. Rep. ESD-TR-75306 June 1975, 1976
4 Lapadula L J. State of the Art in Anomaly Detection and Reaction: [Technical report]. MITRE, Bedford, Massachusetts. 1999
5 Harbron T R. File System Structures and Algorithms. Englewood Cliffs, New Jersey, USA. Prentice Hall, 1988
6 Liu P, Jajodia S, McCollum C D. Intrusion Confinement by isolation in information systems. Journal of Computer Security, 2000, 8(4): 243～279
7 Liu P, Ammann P, Jajodia S. Rewriting histories: recovering from malicious transactions. Distributed and Parallel Databases, 2000, 8(1): 7～40
8 Ammann P, Jajodia S, Liu P. Recovery from malicious transactions. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(5): 1167～1185
9 Panda B, Giordano J. Reconstructing the Database after Electronic Attacks, in Database Security XII: Status and Prospect. S. Jajodia, ed. Kluwer Academic Publishers, 1999. 143～156
10 Ammann P, Jajodia S, Liu P. A Fault Tolerance Approach to Survivability. In: Proc. of the Computer Security, Dependability, and Assurance: From Needs to Solutions, York , England and Washington DC, USA, 1998
11 Jajodia S, Liu P, McCollum C D. Application-Level Isolation to Cope With Malicious Database Users. In: Proc. 14th Annual Computer Security Applications Conf. Phoenix, AZ, 1998. 73～82
12 Fayad A, Jajodia S, McCollum C D. Application-Level Isolation Using Data Inconsistency Detection. In: 15th Annual Computer Security Applications Conf. Phoenix, Arizona, 1999. 119～126

（上接第39页）

4 实验结果和总结

我们的模拟系统由两台笔记本电脑（安装 PCMCIA 接口蓝牙卡）和两台 PC（安装 USB 接口蓝牙卡）所组成。一台 PC 机作为家庭网络平台（HNP Home Network Platform）使用，其他设备模拟服务使用者，或者服务提供者，如日期的换算。我们还用一台 PC 模拟家庭中的老设备—冰箱，使用串口方式连接到 HNP 设备上，并注册自己的控制接口作为服务。

我们使用一台笔记本电脑进行服务调用，无论是以家庭外还是家庭内的调用模式，都成功地设置了模拟冰箱的工作温度和时间，获得了正确的日期换算结果。

采用以 XML 为核心，SOAP 协议和蓝牙 SDP 模块为架构的服务调用机制，最大的特点是：它是基于通信协议的标准化的。这种通信协议的标准化方法，最大限度地保证了现有的各种技术的使用，满足设备生产商的开发要求。浏览器的访问方式可以为用户提供友好的服务界面，并与现有的各种移动设备的要求相符合。同时使用被业界广泛接纳的 XML 和 HTTP 技术保持了蓝牙一贯的高兼容性的特点，并支持用户从 Internet 上使用家庭中的服务，从而实现家庭网络的最终目的：使用户随时随地地使用家庭网络提供的服务。

家庭网络中的服务调用机制还有很多工作需要进行，如多个注册平台之间的相互合作问题，服务调用机制与服务发现协议如何更好地协同工作，服务代理如何更高效地实现复杂对象的调用等。蓝牙-家庭网络中也有很多工作值得研究，如家庭保安系统中如何实现有效的视频传输等。这些都是以后要进行研究的方向。

参 考 文 献

1 Richard G G Ⅲ, Service Advertisement and Discovery. IEEE INTERNET COMPUTING (Sep. & Oct. 2000, 18～26
2 Speification of the Bluetooth System. Available at: http://www.bluetooth.com
3 Arnold K, et al. The Jini Specification. Addison-Wesley Longman, Reading, Mass. , 1999
4 Salutation Architecture Specification. Available online at: http://www.salutation.org/specordr.htm
5 Universal Plug and Play specification v1.0. Available online at http://www.upnp.org/
6 Guttman E. Service Location Protocol: Automatic Discovery of IP Network Services. IEEE Internet Computing, 1999, 3(4): 71～80
7 Miller, Pascoe R. Mapping Salutation Architecture APIs to the Bluetooth Service Discovery Layers. white paper. Available online at: http://www.salutation.org/whitepaper/btoothmaping.pdf.
8 Guttman E, Kempf J. Automatic Discovery of Thin Servers: SLP, Jini and the SLP-Jini Bridge. In: Proc. 25th Ann. Conf. IEEE Industrial Electronics Soc. (IECON 99), IEEE Press, Piscataway, N. J. , 1999
9 房胜. 基于蓝牙技术的家庭网络的设计和实现: [清华大学工学硕士学位论文]. 2002
10 楼颖. 数字化家电网络软件平台 SOPCA 中管理系统的研究: [清华大学工学硕士论文]. 2001