基于状态机的入侵场景重构关键技术研究

冯学伟,王东霞,马国庆,李 津
(北京系统工程研究所,北京 100101)

摘 要:分析了现有的各种安全事件关联算法,提出了一种基于状态机的攻击场景重构技术。基于状态机的攻击场景重构技术将聚类分析和因果分析统一起来对安全事件进行关联处理,为每一种可能发生的攻击场景构建一个状态机,利用状态机来跟踪、记录攻击活动的发展过程,以此来提高关联过程的实时性和准确性。最后通过 DARPA2000 入侵场景测试数据集对所提出的技术进行了分析验证。

关键词:入侵场景重构;聚类分析;因果分析;攻击场景树;关联状态机

中图分类号:TP393 **文献标识码**:A **文章编号**:1674-7720(2010)17-0057-05

Research on the key technology of reconstructing attack scenario based on state machine

FENG Xue Wei, WANG Dong Xia, MA Guo Qing, LI Jin
(Beijing Institute of System Engineering, Beijing 100101, China)

Abstract: The paper analyzes the existing algorithms of security events correlation and proposes an attack scenario reconstruction technology based on state machine. The attack scenario reconstruction technology based on state machine processes security events using the clustering analysis and the cause and effect analysis concurrently, it builds a correlating state machine in the memory for every attack scenario, traces and records the developing of the attack activity. It is more timely and accurately to analyze the security events using this technology. At last, using the DARPA2000 intrusion scenario specific data sets to validate the technology.

Key words: attack scenario reconstruction; clustering analysis; prerequisites and consequences analysis; attack scenario tree; correlating state machine

随着计算机网络的普及和各种网络应用的不断深入,网络空间的安全问题变得越来越突出,已经成为制约网络发展的主要因素之一[1]。对于一个复杂的大规模网络而言,一个简单的攻击指令就会触发 IDS 等安全设备产生数百乃至上千的安全事件,依靠管理员人工分析这些事件无疑是一种灾难。如何从这些零散、庞杂的安全事件中提取出高层警报,重构出入侵场景已经成为当务之急。入侵场景重构技术就是通过对 IDS 等安全设备产生的原始安全事件进行关联、分析,还原出攻击者对整个网络空间的攻击、渗透过程,然后将这种高层的场景信息反映给管理员,将其从繁重的事件分析任务中解放出来。

本文通过分析现有的一些关于安全事件关联分析的研究成果,提出了一种基于状态机的入侵场景重构技术,给出了对应的关联算法以及相关的一些重要概念,最后通过一个原型系统对所提出的重构技术进行了分析、验证。

1 相关工作

攻击活动触发的安全事件之间存在两种关系,一种是并行的冗余关系,另一种是串行的时序因果关系,入侵场景重构的主要内容就是分析各个安全事件之间的这两种关系。

当前对冗余事件的研究主要集中在基于概率的统计分析方面:ALFONSO V[2]首次提出了基于概率聚类技术的安全事件关联分析方法,随后 DEBAR H[3],DAINO[4-5]等也对其进行了研究。基于概率聚类的安全事件关联分析技术通过计算各个安全事件的概率相似度,进而决定新产生的安全事件的聚类归属。通常属于同个聚类的安全事件具有相似的属性,通过选择一个抽象的"元告警"作为该聚类的代表元,以此来达到去冗的效果。

对串行事件的处理主要集中在基于规则的关联分析方面。STEVEN C[6]等提出了一种基于专家系统的攻击场景识别技术，其主要思想是将攻击场景描述为一系列的规则模块，每个规则模块代表着一个攻击场景，安全事件报上来之后和规则模块进行匹配。BENJAMIN M 和 HERVE D 提出了一种基于时序模式识别的攻击场景识别技术[7]，将网络安全事件按照预定义的时序模型进行关联。在基于规则的安全事件关联分析方面，具有里程碑意义的无疑是 PENG Ning 所领导的 TIAA 项目[8]和 ONERA 的 FREDERIC C 所领导的 MIRADOR 项目[9]。

现有的安全事件关联分析技术主要存在以下两个问题：(1)对安全事件的处理要么是进行概率聚类，要么是基于因果规则进行关联分析，很少有研究将两者结合起来处理安全事件。有结合起来的也是人为地将两者割裂，先聚类去冗，然后因果分析，这样做的合理性有待考究。(2)关联分析的实时性差。尤其表现在基于规则的关联分析方面，比如 TIAA 项目，其通常是设定一个时间片，一段时间之后去读取数据库中的安全事件，然后对其进行计算分析。其实质是一个离线系统，而且如果时间片内某一规则的安全事件有遗漏的话，其分析效果会大大降低。

2 基于状态机的入侵场景重构技术

基于状态机的入侵场景重构技术将聚类分析和因果分析统一起来，用于对安全事件进行处理，还原出攻击者的入侵过程。同时基于状态机的安全事件关联分析是一个在线实时的处理过程，这种工作方式可以极大地提高关联过程的时效性。

图 1 为入侵场景重构系统的结构图。整个系统的工作流程如下：首先需要根据专家知识以及参考范例等构建一个静态的入侵场景库，入侵场景库中包含了各种各样的攻击场景，每一个攻击场景都是一个树状结构，用于匹配产生的安全事件。有了入侵场景库之后，预处理模块将各个安全设备产生的事件标准化后提交给关联引擎，关联引擎按照 LRU(最近最少使用)[10]策略在内存中维护着一个状态机队列，队列中的每一个状态机代表着当前网络空间中正在发生的一种攻击场景，这些状态机是入侵场景库中相应攻击场景树的动态表现。安全事件到达之后，关联引擎将其与状态机队列中各状态机的当前状态匹配，如果与某一状态匹配成功，该状态作相应处理，并且将对应状态机调换到队首位置，如果匹配成功的状态是状态机的叶子节点，那么产生一个攻击场景描述事件，并且将该状态机从队列中删除；如果状态机队列中的各状态机都没有可以和安全事件成功匹配的当前状态，那么关联引擎就会去入侵场景库寻找可以和其成功匹配的静态攻击场景树，找到后生成对应的状态机，并且将其插入到队列的队首位置，置该状态机的当前状态为根节点。后续安全事件到达时重复上述关联过程。生成的攻击场景描述事件应该以友好的显示方式呈现给管理员，并且管理员可以手动修改入侵场景库，同时系统还应该对关联过程中的中间数据进行备份。

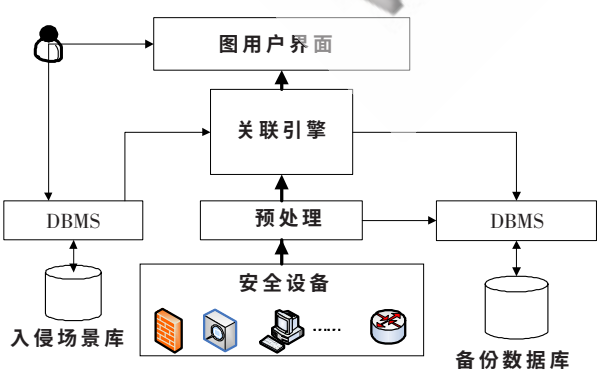

图 1 入侵场景重构系统结构图

2.1 入侵场景库及攻击场景树的定义

定义 1：入侵场景库 *intrusionScenarioBase* 是一个集合 $intrusionScenarioBase=\{AttackScenario\_1, AttackScenario\_2, \cdots, AttackScenario\_i, \cdots, AttackScenario\_n\}$，其中的每一个 *AttackScenario_i* 是由 XML 文件定义的攻击场景树。

入侵场景库定义了网络空间中可能出现的各种各样的攻击场景，如图 2 所示，其可以不断地完善、丰富。入侵场景库是一个预定义的静态结构，其定义来源于专家知识和现有的一些参考范例及管理员的历史积累。场景库中的每一个元素都代表了一种可能的攻击场景，对攻击场景采用 XML 静态文本进行描述。

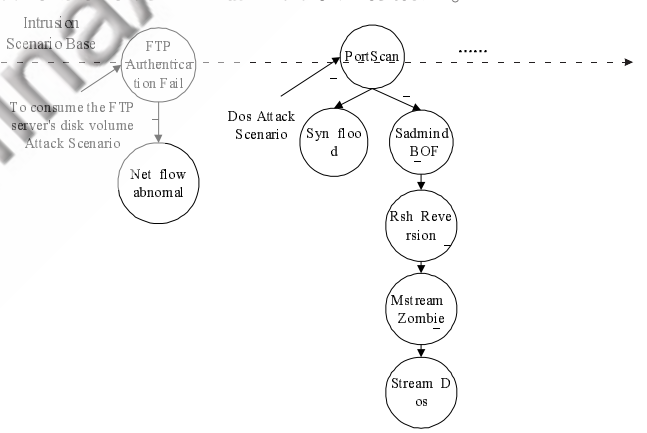

图 2 入侵场景库

定义 2：攻击场景树 *AttackScenario_i* 是对攻击者入侵过程进行描述的一种树型结构，通常由 XML 文档静态定义。场景树中的每一个节点都代表着一个可匹配的规则，规则描述了发生此步攻击的事件的特征。父子节点之间是“与”的关系，表示攻击序列的时序推进，兄弟节点之间是“或”的关系，表示下步攻击的可选方案。树生长的方向就是攻击不断深入的过程。攻击场景树的每一个节点都代表着一条规则，用于匹配安全设备产生的事件。规则之间有着并行的或者递进的关系。

以图 2 中的拒绝服务攻击场景树为例进行说明：

(1)攻击者首先会对目标主机进行端口扫描,查看目标主机上都有哪些服务存活。

(2)找到存活服务后,攻击者有两种攻击方案,一种时直接对目标服务进行 Syn 洪范攻击,致使目标进程拒绝服务。另一种是发现目标主机存在 Sadmind 服务漏洞,攻击者发起 Sadmind 缓冲区溢出攻击,获得目标主机的访问权限。

(3)利用获得的访问权限,攻击者在目标主机上安装用于进行 DDos 攻击的 Mstream 程序。

(4)Mstream 主控端和受控端进行交互,准备进行 DDos 攻击。

(5)被控的傀儡主机一并对目标主机进行拒绝服务攻击。

2.2 关联状态机

定义好各个攻击场景树、形成入侵场景库后,关联引擎将安全设备产生的事件实时地和各个攻击场景进行匹配,还原出攻击者的入侵过程。在匹配时,必须为每一个半匹配的攻击场景维护一个状态机。

定义 3:关联状态机是攻击场景树的一个映射,是关联引擎在内存中维护的一种树型动态结构。每一个状态都是一个十五元组 *state_i*=(*plugin_id*,*plugin_sid*,*src_ip*,*dst_ip*,*src_port*,*dst_port*,*protocal*,*timeout*,*occurrence*,*srcIP_record*,*dstIP_record*,*srcPort_record*,*dstPort_record*,*eventCounter*,*startTime*),前面 9 个属性定义了该状态可以处理的安全事件的特征,其意义与规则中相应属性的意义相同。*srcIP_record*、*dstIP_record*、*srcPort_record*、*dstPort_record* 分别用于记录该状态已经匹配过的安全事件的特征,*eventCounter* 用于记录已经成功匹配的事件个数,*startTime* 用于记录状态生效时间。

关联状态机是一个中间过程,用于实时地跟踪、记录安全事件和攻击场景的匹配过程。*timeout* 和 *occurrence* 是关联状态机的两个核心属性。*timeout* 用于指出关联引擎在每一个状态的最多观察时间,对应于攻击序列中一个攻击步骤的持续时间。*occurrence* 则指出在每一个状态可以关联的安全事件个数,其本质是对一个攻击步骤产生的相同安全事件进行聚类,体现了归并的思想。*timeout* 和 *occurrence* 二者是协同工作的,如果在给定的 *timeout* 时限内关联引擎成功匹配了 *occurrence* 次的安全事件,那么关联状态机就会发生状态迁移,这也对应着攻击序列的渐进。

定义 4:状态机队列是对当前正在发生的各种攻击场景的跟踪、记录,状态机队列中的每一个元素都是一个关联状态机,描述对应攻击场景的实时推进过程。

以 Dos 攻击场景为例说明如何构造一个关联状态机。假设第一个“*PortScan*”安全事件到来,并且状态机队列 *list* 为空。此时关联引擎在入侵场景库中查找哪一个攻击场景的第一条规则要求安全事件为“*PortScan*”类型,发现攻击场景“*Dos Attack Scenario*”满足,此时关联引擎为该攻击场景在内存中建立一个关联状态机“*Dos*”,并且将当前状态 *Curr_State* 设为根节点,如图 3 所示。

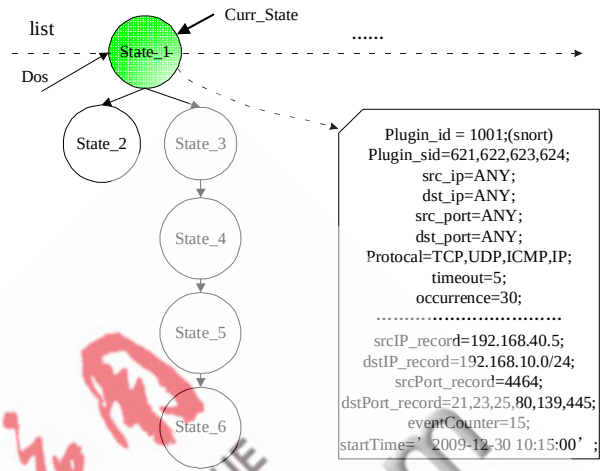

图 3 状态机队列 list 中的 Dos 关联状态机

当前状态为 *State_1*,该状态可以处理任何源到目的端口的扫描类事件,并且对已经成功处理过的事件个数以及这些事件的相关属性进行记录。如果在 5 s 内成功处理的事件个数达到 30 个,那么就会发生状态迁移,当前状态变成 *State_2* 和 *State_3*,如图 4 所示。这一迁移过程对应着 Dos 攻击的第一阶段结束,攻击者可能要进行下一阶段的攻击了。

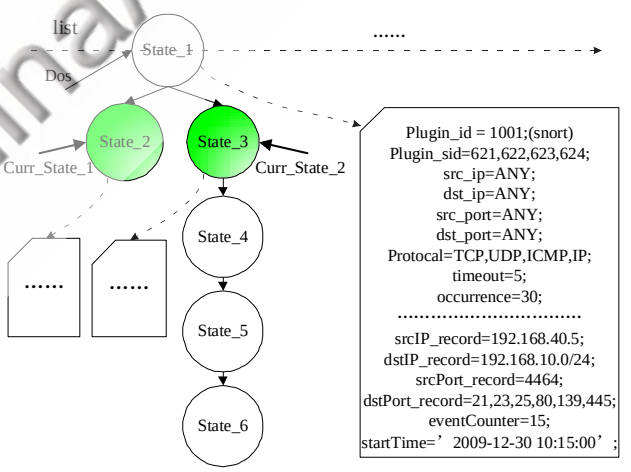

图 4 状态迁移后的 Dos 关联状态机

需要对当前状态成功处理过的安全事件的相关属性进行记录。在记录事件地址时有一个问题:由于一个状态可能处理的安全事件不止一个,而这些安全事件的地址可能相同也可能不同,那么后续状态对先前状态地址引用的时候可能会发生混乱。为了解决这一问题,本文对 IP 地址进行聚析,其好处是可以突出攻击的区域性,可以很直观地反映出攻击的源域和目的域,这在大规模网络中非常有用。对可能出现的不同端口地址利用数组全部保存,如图 3 的 *dstPort_record* 所示。

2.3 关联算法

基于状态机的入侵场景重构系统要求能够对安全设备产生的事件实时处理，重构出入侵者的渗透过程。下面详细介绍基于状态机的安全事件关联算法。

输入：状态机队列 *list*。

输出：攻击场景描述信息 *ScenarioInfo*；更新后的状态机队列 *list*。

```
(1)j=eventRead();   /* 读取预处理后的实时安全事件，如果没有，产生阻塞 */
(2)IF(list==null){
      i=0;
      while(i<intrusionScenarioBase.size){
        IF (Match (j ,AttackScenario_i.firstRule))   /* 在入侵场景库中寻找和 j 一致的攻击场景 */
          break;
        i++;
      }
      IF(i=intrusionScenarioBase.size)
        return;   /* 没有一致攻击场景，缺少预定义的知识 */
      StateMachine=makeStateMachine(AttackScenario_i);
                  /* 为匹配攻击场景构建状态机 */
      StateMachine.CurrentStateSet().add(root);   /* 根节点为当前状态 */
      StateMachine.CurrentStateSet().process(j);   /* 所有当前状态处理事件 j，此时仅仅是根节点状态处理事件 j */
      list.addStateMachine(StateMachine);
    }
(3)ELSE {
      k=0;
      while(k<list.size){
        IF (list.getStateMachine(k).CurrentStateSet().process(j)==true){
                  /* 如果状态机 k 可以处理事件 j */
        IF (the states processing j are leaf nodes of StateMachine_k){   /* 如果能够处理 j 的状态是状态机的叶子节点 */
          ScenarioInfo =generateAttackScenarioInfo (list.getStateMachine(k));
                  /* 生成攻击场景描述信息 */
          delete the leaf states from StateMachine_k.CurrentStateSet();   /* 从当前状态集删除已产生过场景描述信息的叶节点状态 */
        }
        toTheFirstStateMachine (list.getStateMachine (k));   /* 按照 LRU 策略将其置换到 list 的队首位置 */
        break;
        }
        k++;
      }
      IF(k==list.size){
        go to intrusionScenarioBase and do the same thing as list==null except that add the new StateMathine to the head of the list;
      }
    }
(4)goto (1);
```

在关联算法中需要特别注意的是：每一个状态机的当前状态可能不止一个。如在图 3 中 *State_2* 和 *State_3* 合起来构成了当前状态集 *CurrentStateSet*。对安全事件处理的时候，*CurrentStateSet* 中的每一个状态都会参与，如果能成功处理事件，其计数器加 1；如不能，则不做任何动作。如果安全事件和某一状态机当前状态集中的若干状态匹配成功，并且这些匹配成功的状态中有 k 个状态是该状态机的叶子节点，则调用函数 *generateAttackScenarioInfo*() 生成 k 个攻击场景描述信息，每一个攻击场景描述信息详细记录了从该状态机的根节点到相应叶节点的状态变迁路径以及沿途状态的属性信息，随后将这 k 个叶子状态从该状态机的当前状态集 *CurrentStateSet* 中删除。状态机队列中的各个状态机按照 LRU 策略维护，也就是说如果某一个状态机的当前状态集刚成功处理了实时产生的安全事件 j，则将该状态机置换到队首位置，这样做的目的是为了增强关联的实时性。因为根据临近原则，下一安全事件和该状态机成功匹配的可能性最大。

为了实时地维护状态机队列 *list*，必须还有一个单独线程负责监控它，不断查看队列中各个状态机的当前状态，如果在 *timeout* 时限内成功处理的事件个数 *eventCounter* 等于 *occurrence*，则发生状态迁移，并且更新对应状态机的当前状态组 *CurrentStateSet*，如果在 *timeout* 时限临界时 *eventCounter* 仍小于 *occurrence*，表示该状态超时，将其从对应状态机的当前状态组 *CurrentStateSet* 中删除。如果某一状态机的当前状态集 *CurrentStateSet* 为空，则将其从 list 中删除。

3 实验分析

本文实现了一个原型系统，对所提出的入侵场景重构技术进行了分析、验证。通过在网络中回放DARPA2000入侵场景评测数据集[11]对系统进行测试，DARPA2000 数据集是 DARPA 资助 MIT 林肯实验室构造的入侵场景关联评测数据集，其被广泛地应用于验证各种安全事件关联算法的有效性[8,12]。

实验中针对 DDos 攻击定义两种场景树，一种就是图 2 中的 Dos Attack Scenario，另一种是对每步攻击都定义一个场景树，通过这两种方式来验证所提出的基于状态机的入侵场景重构技术。

(1)单步攻击场景检测。林肯实验室给出的DARPA2000数据集总共有 649 787 个数据包，实验对其进行分割，分别得到 5 个攻击步骤各自对应的测试数据集。以第二阶

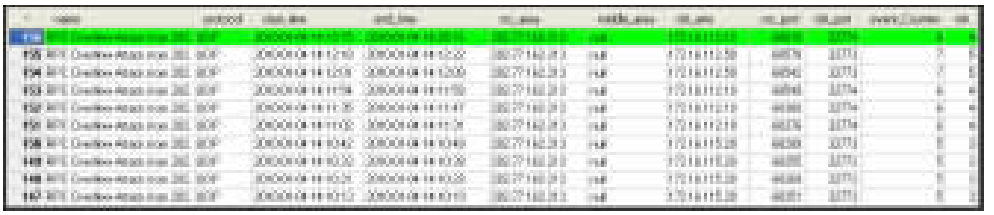

图 5 DDos 攻击中第二步缓冲区溢出攻击的攻击场景描述信息

图 6 DDos 五步攻击场景描述信息

段的缓冲区溢出获取目标主机权限攻击为例进行测试。经过分割后得到该阶段的数据集总共有 12 515 个数据包，利用 TCP-replay 工具在网络中回放该数据集，以 Snort、Snare 和 Ossec 为主要底层安全设备监控网络空间，产生原始安全事件。实验从 2010-1-4 14:08:45 开始，到 14:13:50 时整个回放过程结束，关联引擎对上报上来的原始安全事件进行实时分析处理，最终产生图 5 所示的攻击场景描述信息。

经过分析发现：Snort 等安全设备总共产生了 89 个原始安全事件，关联引擎对这些事件分析处理后产生了 10 个攻击场景描述信息，去冗率达到了 88.76%，关联效率高；回放过程开始于 2010-1-4 14:08:45，在 2010-01-04 14:10:13 时产生了第一个场景描述信息，响应延迟不到 2 min，关联实时性强；攻击场景描述信息中包含了攻击的名称，攻击发生的时间、地点，攻击危害的简单评估值等管理员所关心的安全属性，关联信息丰富。

(2)五步攻击场景完整检测。对于完整的五步攻击，整个数据包的回放过程要持续 3 h 15 min 左右，因此，Dos Attack Scenario 场景树中各个节点的 timeout 属性需要经过仔细推敲才能确定。实验从 2010-1-4 15:10:00 开始回放整个数据包，到 2010-1-4 18:24:13 左右整个回放过程结束。最终关联引擎给出了图 6 所示的 Dos 攻击场景描述信息。

分析图 6 发现，关联引擎给出的攻击场景描述信息较完整地记录了入侵者对网络空间的渗透过程。攻击者 202.77.162.213 在 2010-1-4 16:03:00 左右开始了对目标主机 131.84.1.31 的攻击过程，整个 DDos 攻击过程持续了大概 1 h 20 min，中间用到了傀儡主机 172.16.112.10，172.16.115.20 和 172.16.112.50。DDos 攻击的威胁值达到了 9(最高为 10)，攻击场景图如图 7 所示。

本文提出了一种基于状态机的入侵场景重构技术，将聚类分析和因果分析统一起来对安全事件进行关联分析，还原出攻击者对网络空间的渗透过程，将管理员从琐碎的事件分析任务中解放出来。实验表明，基于状态机的入侵场景重构技术在实际的工作中是有效可行的。下一步工作是继续丰富入侵场景库，并且开发出用户界面，将关联结果更友好地呈现给管理员。

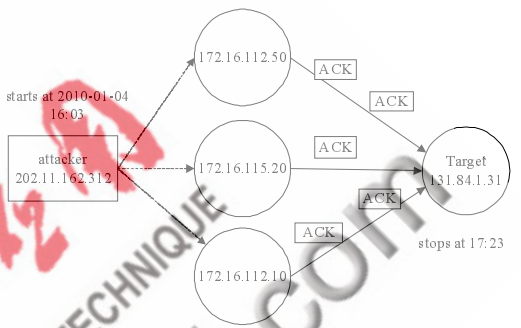

图 7 DDos 攻击场景图

参考文献

[1] EOM Jung-ho, HAN Young-Ju, PARK Seon-Ho. Active cyber attack model for network system's vulnerability assessment [C]. 2008 International Conference on Information Science and Security, 2008.

[2] ALFONSO V, KEITH S. Probabilistic Alert Correlation[C]. Proc. of the 4th International Symposium on Recent Advances in Intrusion Detection. Springer-Verlag, 2001.

[3] DEBAR H, WESPI A. Aggregation and correlation of intrusion detection alerts [C]. Proceeding of the 4th International Symposium on Recent Advances in Intrusion Detection(RAID). 2001.

[4] DAIM O, CUNNINGHAM R K. Building Scenarios from a heterogeneous alert stream[C]. Proceeding of the IEEE SMC Information Assurance Workshop. NY, 2001.

[5] DAIN O, CUNNINGHAM R K. Fusing a heteorgeneous alert stream into scenarios [A]. Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications [C], 2001:1-13.

[6] STEVEN C, ULF L, MARTIN F. Modeling multistep cyber attacks for scenario recognition [C]. Proc of Third DARPA Information Survivability Conference and Exposition. Washington, 2003.

[7] BENJAMIN M, HERVE D. Correlation of intrusion symptoms: an application of chronicles [C]. Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection, Pittsburgh, PA. USA: Springer-Verleg, 2003.

[8] NING P, CUI Y. Techniques and tools for analyzing intru-

sion alerts[J]. ACM Transactions on Information and System Security, 2004,7(2):274-318.

[9] FREDERIC C, ALEXANDRE M. Alert Correlation in a Cooperative Intrusion Detection Framework [C]. Proc. of IEEE Symposium on Security and Privacy, Oakland, California, USA, 2002.

[10] 汤小丹.计算机操作系统(第三版)[M].西安:西安电子科技大学出版社,2007.

[11] 2000 DARPA Intrusion Scenario Specific Data Sets[OL]. [2008-01-24]. http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html.

[12] 韦勇,连一峰,冯登国.基于信息融合的网络安全态势评估模型[J].计算机研究与发展,2009,46(3):353-362.

(收稿日期:2010-03-11)

作者简介:

冯学伟,男,1985年生,硕士研究生,主要研究方向:网络安全、信息融合。

王东霞,女,1975年生,博士,研究员,主要研究方向:网络安全、可信计算。