基于状态转换模型的容侵系统研究*)

崔竞松　王丽娜　张焕国　傅建明　罗　敏
（武汉大学计算机学院　武汉430072）（中国科学院软件研究所计算机科学重点实验室）

The Research of Intrusion Tolerance System Based on State Transition Model

CUI Jing-Song　WANG Li-Na　ZHANG Huan-Guo　FU Jian-Ming　LUO Min
(School of Computer Science, Wuhan University, Wuhan 430072)
(Computer Science Key Lab., Institute of Software, Chinese Academy of Sciences)

Abstract Intrusion tolerance system is a new technology of network security. It can provide acceptable or degraded system service when intrusions occur. In this article, the system's basic function, technique and objective are introduced. A kind of state transition model is discussed. The intrusion tolerance architecture based on state the transition model and several vulnerabilities cases are proposed.

Keywords State transition model, Intrusion tolerance, Network security, Architecture

1　引言

由于互联网的开放性，工作在互联网平台上的各种应用服务器不免会受到来自远端的攻击和入侵。为了减小这些攻击和入侵的可能性，人们设计了各种各样的防火墙(firewall)及其类似产品。实践证明，防火墙能够较有效地抵挡目前已知的大多数简单网络攻击方法。但防火墙(包括类似产品)有以下三个缺点：1. 防火墙通常使用端口对数据包进行过滤。然而这种做法严格限制了客户的行为，使客户能够从应用服务器那里得到的服务难以被扩展，从而大大降低了应用系统的可扩展性和兼容性；2. 防火墙系统的设计和使用通常是与其保护的应用系统无关的。这就使防火墙无法阻止那些针对应用服务器所提供的服务而设计的攻击方案。所以防火墙对于这些真正有效的攻击实际上是无能为力的；3. 即使是对于较为通用的简单的攻击方法，防火墙相应的保护措施的设计和实施往往也是滞后的。而对于新出现的攻击方法，防火墙的保护能力就更加难以让人信赖了。

因此，在经过了防火墙的保护之后，应用服务器仍然有被入侵的可能。于是，一个新的研究方向便应运而生——入侵检测(Intrusion Detection)。入侵检测系统的主要功能是发现将要发生、正在发生或已经发生的入侵行为，并给出相应的报警。入侵检测的方法有很多种，但是总存在漏检或误检，因此要研究容侵系统(ITS)。

作为一个整体，提供服务的应用服务器(或服务器组)不仅仅需要能够抵抗一些简单攻击和发现入侵行为，更关键的是，能够在受到攻击或已经被入侵的情况下，仍能够提供其既定的服务(即使是降级的)，并保持一定的安全底限。

1.1　容侵的作用

容侵的服务对象通常是一些较大规模的应用服务器(或服务器组)。众所周知，大规模应用系统中没有缺陷是几乎不可能的。其中包括了设计缺陷和编程缺陷。人们曾经提出了一些减少缺陷的方法，诸如：软件工程方法、CMM 评价标准、形式化方法等。使用这些方法能够避免大部分设计缺陷和部分编程缺陷。但使用这些方法生产出的产品还需要经过测试和分析。通常在测试阶段能够找出应用系统的大部分遗留的缺陷并加以修正。在应用系统正式投入到使用中时，还需要借助于防火墙来抵抗一些应用系统在设计阶段没有考虑到或没有防范到的攻击，来保护应用系统中的这些缺陷。但最终，还有一些上述手段都没能弥补的应用系统的缺陷，或者弥补起来代价过大的缺陷，就需要容侵系统来保护。

也就是说，容侵系统作为应用系统的最后一道安全防线，将力图使应用系统能提供所要求的服务，必要的时候提供降级服务，并保护服务器上数据的秘密性和完整性。

1.2　容侵的目标

容侵系统的设计目标可以分为四个层次：

(1)保证服务器上数据的真实性、完整性。当网络应用服务器受到入侵时，容侵系统需要保护服务器上的应用数据和系统数据不被非法篡改，或者能够发现已经被篡改的部分，必要时加以恢复。

(2)保证服务器上数据的秘密性。当被入侵者通过非法的途径掌握了对服务器数据的访问权限时，容侵系统需要保证入侵者不能得到明文数据。

(3)保证服务的可用性。当入侵者对应用系统部分服务器的攻击已经成功以后，容侵系统需要使应用系统整体对外仍然能够提供完整服务或降级服务，并力图通过重组和恢复使整体系统回到正常(健康)状态。

(4)保护系统的安全运行。在入侵者尚未攻击，或正在实施攻击但未造成破坏时，容侵系统需要预防、阻止进一步的攻击行为，报警，并力图自动修复系统中的相应缺陷。

以上四个层次是容侵系统的四个主要设计目标。这四个层次是逐层递进的，每后一个都比前一个对系统提供了更强的保护，实现起来也更难。

自1999年起，国际上就开始出现了对容侵技术的系统化研究，目前已经取得了部分成果。国内对于容侵的研究才刚刚起步。

*)国家自然科学基金重点项目(90104005)，国家自然科学基金项目(66973034)。崔竞松　博士生，主要研究方向：网络安全。王丽娜　博士，副教授，主要研究方向：容侵、数字水印、网络安全。张焕国　教授，博士生导师。主要研究领域：信息安全与容错，智能卡技术。罗　敏　博士生，主要研究方向：入侵检测。

2 容侵技术简介

2.1 容侵的策略

容侵系统的策略是指导容侵系统的设计、并决定其运行效果的关键。它与以下因素有关：

(1)容侵的程度　在应用系统受到入侵的时候，容侵系统需要将应用系统保护到的程度。例如：只需要保护服务器上数据的完整性，或者还必须保护数据的秘密性，或者还要提供降级服务，或者还要有自动重组/修复的能力等。

(2)容侵的代价　显然，要实现上述任何一种容侵的程度，都是需要付出相应的代价的。这种代价包括硬件设备上的和软件设计方面的代价，还有效率上的和控制粒度上的代价，还可能包括其它方面的代价。这需要在重要性和代价上进行权衡。

(3)动态配置　对于已经完成的容侵系统还可以在运行的过程中，根据管理员的配置动态调整容侵的策略。但这种动态配置的功能同样也需要付出相应的代价。

2.2 容侵的方法

容侵系统是建立在入侵检测、容错理论和密码理论基础上的系统。它所运用的方法也大多来源于这些方面，它涉及到：

(1)检测与追踪　不言而喻，容侵系统中最前线的就是入侵检测系统。但容侵系统的检测模块与独立的入侵检测系统不同在于：前者的行为往往与容侵系统的其它部分是联动的、紧密耦合的。容侵系统对前者在实时性、准确性方面提出了更高的要求。因此，前者在结构、方法等方面有别于后者。而后者更注重于学习性、自适应性等方面。

追踪是主动式容侵的必要条件。不仅如此，在某些场合追踪还可以为反击提供重要线索。

(2)相关算法　由于容侵系统是建立在容错理论和密码理论的基础上的，因此容侵系统中将会使用到许多与容错和密码相关的算法。例如：冗余(redundance)/屏蔽(mask)算法，检错/纠错算法，门限密码方法，分布式密钥生成算法及其使用方法等。同时，作为网络系统，容侵系统还涉及到相关协议。

(3)重组与恢复　它原本是分布式操作系统所研究的重要问题之一。在容侵系统中，重组与恢复更多地考虑了参与各方的诚实性。在重组和恢复过程中，系统将整体的安全性和继续提供服务的能力放在了极其重要的位置上。

3 基于状态转换模型的容侵机制

3.1 模型的定义

由于容侵系统可以保护的对象是多种多样的，所以每个容侵系统所采用的系统框架、容侵策略、安全算法都不尽相同。因此，为了便于抽象地研究容侵系统的工作流程，我们引入了状态转换模型[1]，从宏观上描述了容侵系统的工作流程。我们认为，容侵系统在抵抗入侵的过程中，系统的状态是在一个状态转换图中迁移的，如图1所示。

3.2 容侵机制

我们认为，在上述状态转换图中，一个容侵系统的容侵机制是这样实施的：

(1)状态 G：我们假设 COTS (Commercial-Off-The-Shelf)服务器在投入运行的时候是处于正常状态 G(Good)的。但我们也同时认识到：COTS 服务器通常是存在着缺陷的。也就是说，当攻击者掌握了这些缺陷，这些服务器是有可能受到攻击的。

(2)状态 V：当攻击者准备对服务器进行攻击时，通常会有一些前奏。例如：扫描端口、上载恶意数据等等。这个时候，系统将进入脆弱状态 V(Vulnerable)。此时，攻击者还没有对应用和服务造成伤害。如果此时这种攻击行为的前奏被检测到了，则系统可以试图通过相应的预防措施返回 G 状态。例如：暂时封锁该地址、删除恶意代码等。

(3)状态 A：如果攻击行为没有任何前奏，或者前奏没有被检测到，则系统随时有可能被攻击。一旦攻击者发动攻击，则系统进入被攻击状态 A(Attack)。在 A 状态下，应用服务器的某些部分或者某些功能已经被损害。这种损害可能是静态的、一次性的(例如：破坏数据等)，也有可能是动态的、有延续性的(例如：创建了恶意进程等)。

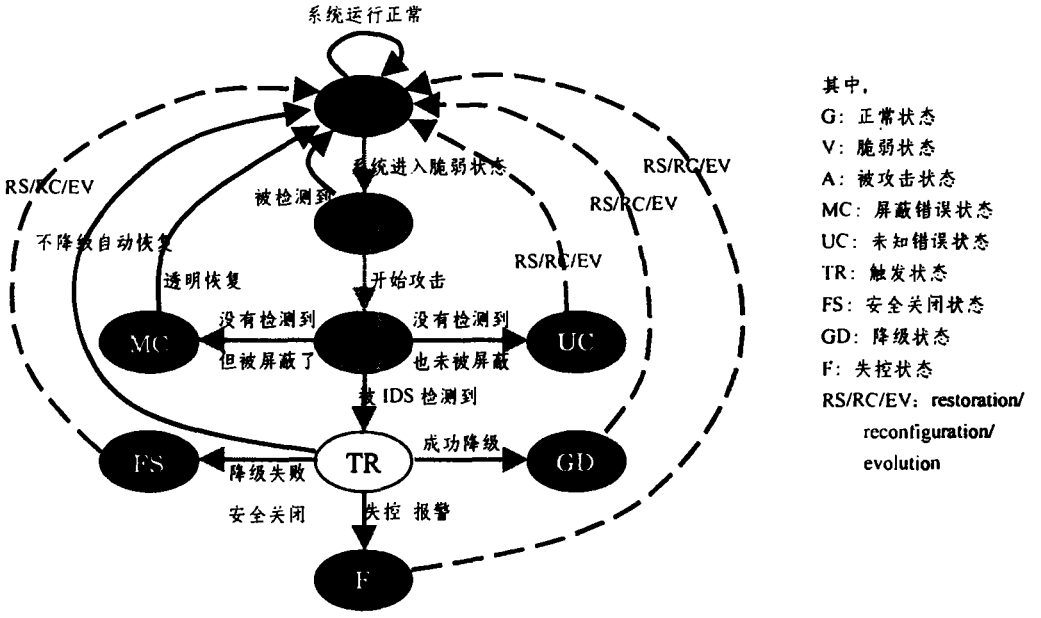

图1　状态转换图

(4)状态 UC：如果入侵没有被检测到，也没有采取任何措施加以控制和消除，则系统进入了未知错误状态 UC (Undetected Compromised)。在这个状态下，系统将带着已经造成的损害继续运行，但其提供的服务和系统的安全性将无

法得到保证。这是一种非常危险的状态。从这种状态恢复到正常状态G的唯一办法就是手动恢复(下文将详细描述)。

(5)状态MC:如果入侵没有被检测到,但在系统的设计中已经准备了一些容错措施对这种损害加以控制和消除,则系统进入屏蔽错误状态MC(Masked Compromised)。例如:系统中已经存在了3选2的容错机制,或其他屏蔽(Mask)机制,则系统整体对外仍然保持着正常运行。而此时,如果系统中还设计有纠错机制,则系统可以自动修复被破坏的数据和资源,以透明的方式恢复系统,返回G状态。

(6)状态TR:如果入侵检测系统IDS成功地检测到了入侵行为,进入触发状态TR(Triage),则容侵系统将获得更多的主动权。首先,系统可以尝试自动恢复。例如:终止来历不明的进程等。如果成功,则系统将回到G状态。

(7)状态GD:如果自动恢复不成功,则还可以尝试通过自动重新配置(Auto Reconfiguration),提供降级服务,进入降级状态GD(Graceful Degradation)。降级服务是指:降低所提供的服务的种类、规模、性能等,以换取保证正常提供某些关键的、基本的服务。例如:限制处理请求的个数,暂停响应低优先级的请求,关闭某些应用端口等,以保证重要用户和管理员的正常操作。重新恢复到正常状态需要人工干预。

(8)状态FS:如果连降级服务也无法提供,为了保证服务器数据的秘密性、完整性,可以立即关闭所有的服务,并将服务器安全关闭,进入安全关闭状态FS(Fail Secure),避免更大的损失。此时,需要管理员手动对系统进行修复,才能重新启动服务。

(9)状态F:如果服务器完全失控,无法自动关闭,则属于失控状态F(Fail),应立即报警,请求管理员处理。

状态转换模型描述了一个一般化的容侵系统在抵抗入侵时可能发生的事件和所处的状态。其中,当系统处于非正常状态时,有时可以自动恢复到G状态,有时需要手动恢复。

3.3 手动恢复

手动恢复是指需要管理员干预的恢复方式。管理员可以采用以下手段进行恢复:

1. 恢复被破坏的数据和资源,并启动相应的服务。(restoration)

2. 放弃受损的部分,启动备用方案,重新组织系统内的资源,继续提供服务。(reconfiguration)

3. 寻找并修复系统中的相应缺陷,杜绝攻击者再次攻击的可能。(evolution)

我们可以通过下面这个实例来了解状态转换图是如何描述容侵系统的工作的。

4 容侵实例

4.1 实例一

有一种针对Sun Java Web Server的攻击方法,它可以以管理员权限在服务器上执行一段攻击者设计的代码。方法如下:该Web Server上有一个提供BBS功能的board.html,攻击者将自己设计的JSP代码作为一个消息,通过/examples/applications/bboard/bboard_frames.html,连接进入board.html,然后通过远程调用http://target:9090/servlet/com.sun.server.http.pagecompile.jsp92.JspServlet/board.html,在服务器上以管理员权限激活这一段JSP代码。

我们用状态转换模型的方法来分析容侵系统在对抗这种攻击时的情形:

(1)状态G:首先当Sun Java Web Server启动时,我们认为系统处于G状态。

(2)状态V:当攻击者将JSP代码作为消息,连接进入board.html时,系统进入了V状态。如果容侵系统发现了BBS消息中含有来历不明的JSP代码,可以立即删除该JSP代码,返回G状态。

(3)状态A:如果容侵系统未对BBS消息进行此项检查,而攻击者成功地激活了上载的恶意代码,则系统进入了A状态。

(4)状态UC:如果被启动的JSP程序未被容侵系统检测到,服务器也没有对其上运行的JSP的权限进行严格限制,则系统进入UC状态。此时,只能依靠管理员发现服务器的异常行为,再做出相应处理。

(5)状态MC:而如果服务器对其上运行的JSP的权限已经做出了严格限制,则这段恶意代码的破坏将被Java虚拟机屏蔽,进入MC状态,直至该代码自动终止,或超时被虚拟机终止,返回G状态。

(6)状态TR:如果恶意JSP程序的运行被容侵系统检测到了,则系统进入了TR状态。虚拟机可以立即终止该程序,使系统返回到G状态。

(7)状态GD:如果终止该程序失败,系统还可以尝试提供降级服务。例如:终止BBS服务功能,仅保留Web服务,进入GD状态。然后,等待管理员修复系统,再返回G状态。

(8)状态FS:如果恶意JSP程序已经造成了严重损害,无法继续提供Web服务,则服务器可以尝试进入FS状态,关闭所有的网络服务,关闭Web服务器。等待管理员修复系统,再返回G状态。

(9)状态F:如果服务器无法自动关闭,不能进行自我保护,则应立即报警,请求管理员处理,进入F状态。

在这个实例中,我们可以看到状态转换模型有能力准确地描述这个容侵实例的工作流程。

4.2 实例二

有一种对Windows2000 Server IIS的攻击方法如下:当攻击者向IIS服务器提出GET请求,并携带参数/c/winnt/system32/cmd.exe /c...(或者其它类似的参数),要求返回CGI运行结果时,如果没有安全地设置IIS服务器的CGI启动目录,IIS服务器就会运行cmd.exe并执行所带参数中的命令,将结果返回给攻击者。在攻击时,攻击者会首先用dir之类的命令测试IIS上是否有这样的安全漏洞。如果有,攻击者会用ftp命令将恶意代码下载到被攻击的主机上,最后用cmd.exe在管理员权限下激活恶意代码。

我们用状态转换模型的方法来分析容侵系统在对抗这种攻击时的情形:

(1)状态G:首先当IIS 5.0启动时,我们认为系统处于G状态。

(2)状态V:当攻击者使用dir命令测试并发现服务器上有这个安全漏洞后,使用ftp命令将恶意代码下载到服务器上时,系统进入了V状态。如果容侵系统发现了非法的CGI请求时,可以暂时封锁该IP地址;当发现有未授权的下载行为时,可以终止下载并删除所下载的数据,返回G状态。

(3)状态A:如果容侵系统未发现攻击者的上述操作,而攻击者利用cmd.exe激活了上载的恶意代码,创建了攻击者的进程,则系统进入了A状态。

(4)状态UC:如果被启动的攻击者进程未被容侵系统检测到,且攻击者进程的权限不受控制,则系统进入UC状态。此时,只能依靠管理员发现服务器的异常行为,再作出相应处理。

(5)状态MC:如果服务器能够对攻击者进程的权限进行

有效控制，则这段恶意代码的破坏将屏蔽，进入 MC 状态，直至该代码自动终止，返回 G 状态(但目前尚未发现能够对拥有管理员权限的恶意进程进行有效控制的方法)。

(6)状态 TR：如果攻击者进程的运行被容侵系统检测到了，则系统进入了 TR 状态。任务管理器可以立即终止该程序，使系统返回到 G 状态。

(7)状态 GD：如果终止该程序失败，系统还可以尝试提供降级服务。降级的方法可以视攻击者进程所造成的破坏而定，进入 GD 状态。然后，等待管理员修复系统，再返回 G 状态。

(8)状态 FS：如果攻击者进程已经造成了严重损害，无法继续提供 Internet 服务，则服务器可以尝试进入 FS 状态，关闭所有的网络服务，关闭 IIS 服务器。等待管理员修复系统，再返回 G 状态。

(9)状态 F：如果服务器无法自动关闭，不能进行自我保护，则应立即报警，请求管理员处理，进入 F 状态。

在这个实例中，我们可以看到尽管 MC 状态的实施方法尚未成熟，但从总体上，状态转换模型描述这个容侵实例的工作流程。

结论 本文在介绍了 Firewall、IDS 和 IT 之后，引入了状态转换模型，对容侵系统的工作流程进行抽象的描述。本文对状态转换图中的状态进行了详细的解释，并通过实例说明了容侵系统在抵抗入侵的时候，系统状态在图中的迁移过程和相应方法。

在一般的容侵实例中，未必状态转换模型中的所有状态、事件和处理方法都有所对应。在有些情况下可能只存在状态转换图中的某些状态，也有可能某些恢复方法找不到对应的实施手段。但从整体上讲，状态转换模型对于描述一般容侵系统的工作流程，是很有代表性的。

在状态转换模型中，我们需要进一步研究的课题还有：对协同/并行攻击的容侵模型的建模与分析、容侵系统与 COTS 服务器的组织结构、对于特定环境的重组和恢复方法、入侵检测方法等。

参 考 文 献

1 Characterizing Intrusion Tolerant Systems Using A State Transition Model. http://www.anr.mcnc.org/projects/SITAR/papers/darpa00.pdf, 2000
2 Bugtraq 1600. http://www.securityfocus.com/bid/1600,2000
3 Auscert advisory aa-2000.02. http://ftp.auscert.org/pub/auscert/advisory/AA-2000.02, 2000
4 Du W, Mathur A P. Vulnerability testing of software system using fault injection: [Technical Report Coast TR-98-02]. Department of Computer Science, Purdue University, 1998
5 MCNC, D. University. Sitar : A scalable intrusion tolerant architecture for distributed services:[Technical report, Research Proposal to DARPA BAA-00-15]. 2000
6 Lee P A, Anderson T. Fault Tolerance: Principles and Practice. Springer Verlag, 1990
7 Amoroso E G. Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. Intrusion. Net Books, 1999
8 Krsul I, Spa®ord E H, Tripunitara M V. Computer vulnerability analysis: [Technical Report Coast TR 98-07]. Department of Computer Science, Purdue University, 1998
9 Ellison R J, et al. Survivability: Protecting your critical systems. IEEE Internet Computing, 1999,3(6):55～63
10 Northcutt S, Novak J. Network Intrusion Detection: An Analysts' Handbook. New Riders, Sep. 2000

(上接第50页)

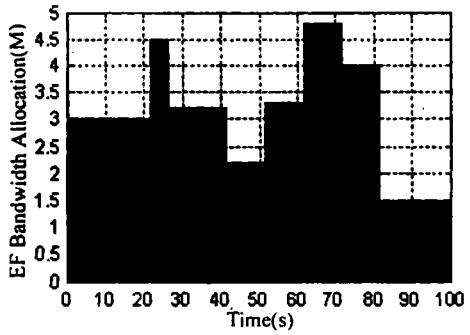

图8 EF 类带宽资源0时刻的分配情况

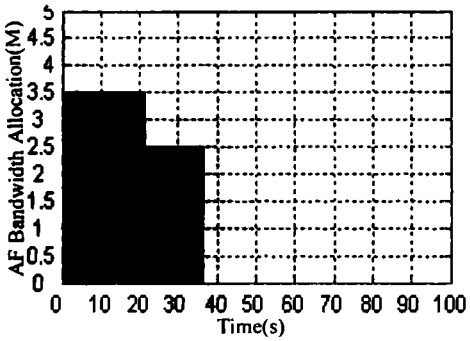

图9 AF 类带宽资源0时刻的分配情况

从图8中可以看到，在试验时间内，S_1和S_9对 EF 类带宽资源的最高需求为4.8M。如果采用静态的带宽分配方式，则整个时间段内带宽分配都必须是4.8M，对用户而言，将浪费大约30%的带宽。因此，动态的资源分配方式和用户的利益是相一致的。

结论 要使区分服务模型得到实际的应用，不光要有良好的机制来实现 PHB，也需要动态、灵活的资源分配和管理体制，满足不同用户的 QoS 要求。本文提出了基于 BB 的动态带宽资源分配框架，给出其原型设计。仿真试验表明，该方案能满足动态分配带宽资源的要求，并能实现资源预留。但是，这只是我们工作的第一步。我们将在资源预留下如何接纳控制算法的高效实现、BB 间通信协议的完整设计、BB 的可伸缩性等一系列问题上做进一步的研究。

参 考 文 献

1 lake S, et al. An Architecture for Differentiated Services. RFC 2475, Dec. 1998
2 Teitelbaum S, et al. Internet2 Qbone: Building a Testbed for Differentiated Services. IEEE Networks, Sep./Oct. 1999
3 Reichmeyer F, et al. A Two-Tier Resource Management Model for Differentiated Services Networks. Internet Draft, draft-rotzy-2-tier-management-00.txt, Nov. 1998
4 Nichols K, Jacobson V, Zhang L. A Two-bit Differentiated Services Architecture for the Internet. RFC2638, Apr. 1999
5 Fall K, et al. The ns Manual. http://www.isi.edu/nsnam/ns-docmentation, April 2001
6 Pieda P, et al. A Network Simulator. Differentiated Services Implementation. Open IP, Nortel Networks, July 2000
7 Jacobson V, Nichols K, Poduri K. An Expedited Forwarding PHB. RFC2598, June 1999
8 Heinanen J, Baker F, Weiss W, Wroclawski J. Assured Forwarding PHB Group. RFC2597, June 1999