信息系统的可存活性*)

胡方炜　李千目　许满武
(南京大学软件新技术国家重点实验室　南京 210093)

摘　要　可存活性是用来表明系统在面对蓄意攻击、故障失效或偶发事故时仍能完成其任务的能力。可存活性要求系统具有四个关键性质:抵抗性、识别性、恢复性、适应和演化性质。目前对于可存活性的研究可以分为分析和实现两个方面。可存活性的分析一般通过建模的方法,对原有系统的可存活性进行度量,找到系统的薄弱环节。可存活性的实现一般通过体系结构的设计改进或者重新配置的方法,来提高系统的可存活能力。无论是分析还是实现的研究,方法多样,但是却有着很大的局限性。

关键词　可存活性,可存活性的分析,可存活性的实现

Survivability of Information Systems

HU Fang-wei　LI Qian-mu　XU Man-wu
(State Key Laboratory of Computer Software and New Technology, Nanjing University, Nanjing210093, China)

Abstract　Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. Survivability requires the system to have four key properties: resistance, recognition, recovery, adaptation and evolution. At present, the study on the survivability can be divided into two aspects, analysis and realization. Survivability analysis generally measures the system's survivability through modeling, and then finds out the weak parts of the system. Survivability realization is to improve the system's survivability through the design, improvement of the system or reconfiguration. There are lots of methods in both the analysis and realization research, but they have some limits.

Keywords　Survivability, Survivability analysis, Survivability realization

1　引言

信息技术已经深入到社会生活的方方面面,虽然给人们的生活带来了方便,但是由于网络的复杂性,系统规模和速度的迅速增长,系统之间依赖性的加强,系统自身的安全性受到了很大的威胁。与人们日常生活息息相关的电信、金融、医疗等基础设施系统亦在其列。

传统的安全技术研究着重于防御,即检测并阻止系统可能受到的入侵威胁。这种技术对系统做了很多的加固(hardening),可是入侵威胁防不胜防。经验表明信息系统很难保证不受到攻击,在受到攻击时亦不能保证其绝对安全,即现阶段制造出一个绝对安全的系统是不可行的[1]。

网络系统已经发展到大规模、分布式的无限网络,处在无限网络下的信息系统更易受到攻击。随着第三方商品软件(Commercial Off-the-Shelf)和公共构件的应用发展,有关系统内核的大量信息公开传播,入侵技术更是发展迅速。如何在开放的复杂环境下,面临攻击时仍然能够保持系统的运转,即系统的可存活性,成为当前系统安全性研究的重点。

2　可存活性的定义

到目前为止,可存活性还没有统一的定义。不同的研究工作者、不同的领域专家对于可存活性的理解亦是不尽一致的。

目前比较受业界认同的是由卡内基梅隆大学的 R. Ellison 等人提出的可存活性定义:可存活性是用来表明系统在面对蓄意攻击、故障失效或偶发事故时仍能完成其任务并及时恢复整个服务的能力[2]。

系统的可存活性与系统的其他很多性质相互关联,如可靠性(reliability)、安全性(security)。文献[3]总结了 20 个与之相关的性质,并进行了解释。可存活性与可靠性、安全性等相关,但并不等同。这些性质与可存活性不是包含与被包含的关系,也不能说可存活性是这些性质的组合。它们之间互有关联,但是研究侧重点却又不尽相同。

可靠性是指在一给定时间内,系统不间断提供服务的能力。它更适合用来评估系统对灾难的防御能力。安全性研究重点在于抵御入侵,即入侵尚未成功侵入系统之前系统自身的防护能力。可存活性的研究是基于这些相关性质的研究,但同时又引入了新的概念和原理。可存活性强调的是入侵成功或者灾难发生之后,系统能够继续提供服务,以及条件状况改善时系统能够自动恢复的能力。

3　可存活性系统的特征

可存活性系统的一个关键特征就是在攻击(attack)、失败(failure)或意外(accident)发生时仍然能够提供关键服务。这就要求即使系统的一个重要部分失灵时,系统仍保有此能力。而且,这种能力不应仅仅依赖于某一专门的信息资源、计算或

*)基金项目:中国博士后科学基金,江苏省自然科学基金。**胡方炜**　硕士研究生,主要研究领域为软件方法学、信息安全;**李千目**　博士后,主要研究领域为信息安全、网络性能分析;**许满武**　教授,博士生导师,主要研究领域为软件方法学、信息安全。

通信连接。

为了保有此种能力，可存活性系统应该具有四个关键性质：抵抗性（Resistance），识别性（Recognition），恢复性（Recovery），适应和演化性质（Adaptation and Evolution）。

四个性质的具体描述如表1所示。

表1 可存活性系统的关键性质

关键属性	定义	相关技术举例
抵抗性	指抵御和阻止攻击的能力	防火墙；用户认证；加密多样化技术
识别性	指识别出攻击或攻击之前的刺探（probing）的能力	入侵检测技术；自省（self-aware）内核完整性核查；周期性 trust 维护；Blackbox Report
恢复性	指入侵后还原服务的能力，提高入侵抵制和识别的能力	容错技术；关键服务复制；软、硬件备份技术
适应和演化	基于入侵获得的知识来提高系统的可存活性，减少未来攻击的有效性	动态更新技术；软、硬件版本升级

4 可存活性的研究

4.1 研究分类

目前可存活性的研究根据不同的标准，可以有不同的分类方法。

根据研究的重点，分为对网络连接的存活性研究和对系统自身的存活性研究。网络连接的存活性主要研究当遭遇自然灾害而出现网络连接故障时，该网络不同节点之间保持通信的能力。系统自身的可存活性是指当入侵成功后，系统能继续执行其关键任务的能力。

根据研究的方向，分为系统可存活性的分析和系统可存活性的实现两种方向。系统可存活性的分析是指从定量和定性的角度，对当前系统的可存活能力进行度量；系统可存活性的实现是从设计系统的体系结构或者改进系统算法的角度，来提高系统的可存活能力。

根据研究的领域，可以划分为对银行、医疗、电信等基础设施系统的可存活性研究；对军事防御系统的可存活性研究等不同领域范围的研究。

4.2 系统可存活性的分析

可存活性的分析是用来量化当前系统在攻击中存活的能力并识别系统中易受攻击的组件。

4.2.1 评判可存活能力的标准

可存活性的评判标准正如可存活性的定义一样，没有一个统一的标准。不同研究工作者、不同研究领域根据需要采用不同的标准分析系统的可存活性。

就目前来说，评判标准大体上可归为三类：连通性（connectivity）；性能（performance）；其他性质或者代价（cost）的函数[3]。连通性主要用在对网络连接可存活性的分析中。性能包含很多方面，如识别力（recognition）、抵抗力（resistance）等等，可用于系统自身可存活性的分析。文献[4]便是采用"3R"——Resistance，Recognition，Recovery 作为标准分析；文献[5]中则采用 attack-potential 来分析。其他性质或者代价的函数，如文献[6]中利用函数 Fm 来度量，Fm 的定义根据需要可以改变。

4.2.2 分析可存活性的方法

对可存活能力的分析方法一般采用建模（model）的方法，分为两步：先对攻击（attack）进行分析，然后对系统的存活性进行评估。

对攻击的分析亦可有多种选择，常用的方法有攻击树（attack tree）[6]、图论方法[7]。

基于攻击分析方法的不同，可以选择不同的可存活性评估模型。常用的模型有马尔可夫链（Markov chain）[8,9]、有限状态机（Finite state machine）[9]、图（Graph）[7,10]、基于状态转换（State transmission based）[5]的模型和差错引入（Fault injection）模型[11]。

文献[6]中以攻击树来建立攻击模型，对于信息系统首先建立一棵攻击树，然后对该树的叶子结点（代表每一个系统组件）进行攻击模拟，计算出每个组件易受攻击的能力，即可存活性能力，从而评估出系统整体的可存活性。文献[7]中用图来表示网络系统，并利用数学的方法来分析图，得到单个结点、两个结点之间以及整个网络系统的可存活性分析。文献[8]则用 CSL（Continuous Stochastic Logic）来描述可存活性，用马尔可夫链建模分析系统可存活能力的大小[11]。引入各种 fault 到系统中，通过反复测试，度量系统的存活能力，并找出系统的薄弱环节。

此外还有很多其他方法，如文献[4]中采用了层次式的方法来分析系统的可存活性。

4.2.3 分析方法的局限性

现在的研究多局限于对某一种系统和某一类情况的研究，只是个性化的研究。分析方法的局限性的一个重要表现为分析方法的多种多样。

分析方法的多种多样虽然表明可以从各个方面考察研究系统的可存活性，但是另一方面也说明目前很难找到一个统一的分析方法，适用于各种情况下对不同类型的系统进行可存活性分析。此外，在分析攻击（attack）时，也只是对于某一类型的攻击进行建模且时常带有附加条件限制和假设。这些均表明现有的研究停留在个性化研究的阶段。

分析方法的局限性的另一个表现为缺乏实际应用性。许多分析方法理论上看似十分合理，但是很难应用到实际的系统可存活性分析中。

4.3 系统可存活性的实现

可存活性的实现就是通过改进系统使得系统具有可存活的能力。

目前可存活性实现的方法主要集中在两个方面：体系结构的设计和重配置（reconfiguration）。

4.3.1 体系结构的设计与改进

体系结构的设计和改进可以提高系统的可存活性。针对不同的应用，可以设计不同的体系结构。

文献[12]设计了一个两层结构，低层用于判断系统是否处于受攻击状态，当攻击成功时，使用高层处理不同类型的攻击。文献[13]则在系统体系结构上设计了一个 workflow 层，该层用于模拟触发事件的效果；如若该事件恰好匹配上某种攻击，则拒绝该事件并阻止该攻击产生的错误蔓延。文献[14,15]研究了在自治无人系统中，借助感知学、动物学方面的知识设计了一个体系框架来实现系统的自治性和可存活性。

虽然体系结构的设计可以有效地提高系统的可存活力，使得系统面对攻击时仍能继续提供服务，但是体系结构的设计研究花费巨大，而且实现困难，不易验证。

4.3.2 重配置（Reconfiguration）

重配置亦可用来实现系统的可存活性。重配置又可细分为两类。

一种是配置冗余，即配置多样化，从多种配置中选出一种具有最大可存活性的配置。文献[16]讲述了通过依赖关系矩阵来选择，获得各种服务之间依赖关系的最大可存活性，从而确定服务之间的依赖关系。文献[17]通过设置服务复本的多少及位置，实现当服务受到攻击后，复本可以继续提供服务，而且可重新配置，使得原本存放服务 A 的复本空间根据需要存放服务 B 的复本。

另外一种是通过重新分配资源给关键任务来保证攻击成功后，系统仍然能够完成关键的服务。这种方法首先要确定哪些是关键任务，哪些为非关键任务。文献[18]提供了 ERAS 的方法，在系统受到攻击后，通过剥夺非关键任务的资源，将其重新分配给关键任务来实现系统的可存活性。文献[19]则通过 Cactus 方法，使得系统在运行时刻可以动态地改变自身的行为和配置来实现可存活性。

4.3.3 *实现方法的局限性*

现有的方法无论是设计体系结构还是重新配置，都默认了一个前提：攻击是事先可知的，或者说是事先假定好的。系统的可存活能力只在某一种或者几种类型的攻击发生时有效，对于突然出现的(emerging)、事先不清楚的攻击无能为力。

此外，同分析方法类似，方法多而杂，且实用性不足。

结束语 近年来，针对系统的蓄意攻击给国家资源和基础设施(如医疗、银行等)带来的损失愈来愈大，系统的可存活性变得愈发重要。什么是可存活性的本质以及怎样描述可存活性，是可存活性研究领域中两个最基本的问题。如何分析现有系统的可存活性，以及如何提高系统的可存活性成为当前研究的两大课题。

由于系统的可存活性至今尚无一个统一的定义，这两大课题的发展面临着巨大的挑战——方法种类繁多，但具有代表性的经典方法不多；研究重在理论，实用性不强；研究偏重某一方面，全局性不足。将来的研究应着手从多方面考虑系统可存活性的分析与实现，少些假设，多些实用。

参 考 文 献

[1] Kyamakya K. Security and Survivability of Distributed Systems: an Overview // IEEE MILCOM 2000. Los Angels, California, 2000:449-454

[2] Ellison R, Fisher D, Linger R, et al. Survivable Network Systems: An Emerging Discipline [R]. CMU/SEI-97-TR-013. 1997

[3] Westmark V R. A Definition of Information System Survivability // Proceedings of the 37th Hawaii International Conference on System Sciences. Hawaii, 2004

[4] Lin Xuegang, Xu Rongsheng, Zhu Miao liang. Survivability Computation of Networked Information Systems. CIS, 2005, Part II:407-414

[5] McDermott J. Attack-Potential-Based Survivability Modeling for High-Consequence Systems // Proceeding of Third IEEE International Workshop on Information Assurance. College Park, Maryland, 2005

[6] Fung C, et al. Survivability Analysis of Distributed Systems Using Attack Tree Methodology // Military Communications Conference. IEEE, 2005:17-20

[7] Yi Xun, Zhang Yanchun. Survivability of Information System. *Information, Communications and Signal Processing*, 2005: 1551-1555

[8] Cloth L, Haverkort B R. Model Checking for Survivability. Quantitative Evaluation of Systems, 2005:145-154

[9] Jha S, Wing J, Linger R, et al. Survivability Analysis of Network Specifications // International Conference on Dependable Systems and Networks. New York, USA, 2000

[10] Krings A W, Azadmanesh M H. A Graph Based *Model* for Survivability Analysis: Technical Report, UI-CS-TR-02-024. 2004

[11] Voas J M, Ghosh A K. Software Fault Injection for Survivability // DARPA Information Survivability Conference and Exposition. 2000:338-346

[12] Harrison W S, Krings A W, Hanebutte N. On the Performance of a Survivability Architecture for Networked Computing Systems. System Sciences, 2002:2534-2542

[13] Xiao Kun, Chen Niannen, Ren Shangping, et al. A Workflow-based Non-intrusive Approach for Enhancing the Survivability of Critical Infrastructures in Cyber Environment. Software Engineering for Secure Systems, 2007:4-10

[14] Quek B K, Ibanez-Guzman J, Lim K W. A Survivability Framework for the Development of Autonomous Unmanned Systems. Control, Automation, Robotics and Vision, 2006:1-6

[15] Quek B K, Ibanez-Guzman J, Lim K W. Attaining Operational *Survivability in an Autonomous Unmanned Ground Surveillance* Vehicle // IEEE Industrial Electronics, IECON. 2006:3969-3974

[16] Lu Tun, Gu Ning. Survivability-Aware Configuration Management of Service-Oriented System Based on Service Dependency. Theoretical Aspects of Software Engineering, 2007:421-432

[17] Wells D, Ford S, Langworthy D, et al. Software Survivability // DARPA Information Survivability Conference and Exposition. vol. 2, 2000:241-255

[18] Wang Jian, Wang Huiqiang, Zhao Gaosheng. ERAS—An Emergency Response Algorithm for Survivability of Critical Services. Computer and Computational Sciences, 2006, 2:97-100

[19] Hiltunen M A, Schlichting R D, Ugarte C A, et al. Survivability through Customization and Adaptability: The Cactus Approach // DARPA Information Survivability Conference and Exposition. vol. 1, 2000:294-307

(上接第 252 页)

要最优整像素运动矢量周围点的匹配误差，可以与任意整像素搜索算法配合使用，更易于推广，具有很强的实用性。

参 考 文 献

[1] AVS 工作组. 信息技术——先进音视频编码，第七部分：移动视频(报批稿)[S]. 2006

[2] Tourapis A M, Au O C, Liou M L. Fast block matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST). ISO/ IEC J TC1/ SC29/ WG11 MPEG99/ m5866, Noordwijkerhout, the Netherland, Mar. 2000

[3] Tourapis A M. Enhanced predictive zonal search for single and multiple frame motion estimation // Proceedings of Visual Com2 munications and Image Processing 2002 (VCIP-2002). San Jose, CA, January 2002:1069-1079

[4] Zhou Z, Sun M T, Hsu Y F. Fast variable block-size motionestimation algorithms based on merge and split procedures for H. 264/MPEG-4 AVC[C] // IEEE International Symposium on Circuits and Systems. Vancouver, Canada, 2004:725-728

[5] Chen Z B, Zhou P, He Y. Fast integer pel and fractional pelmotion estimation for JVT[Z]. fttp://standards. po lycom. com/ imtc_jvtexperts/2002_12_Awaji/JVT-F017. zip

[6] Chen Z B, Du C, Wang J H, et al. PPFPS- A paraboloid prediction based fractional pixel search strategy for H. 26L // Proc. of ISCAS 2002. May 2002:9-12

[7] Yang Libo, Yu Keman, Li Jiang. Prediction - based directional fractional pixel motion estimation for H. 264 video coding [C] // IEEE International Conference on Acoustics, Speech and Signal Processing. Philadelphia, PA, 2005:901-904