基于组件的主动网络安全机制研究*

周恒琳, 李增智, 武亚强, 廖志刚
(西安交通大学 电信学院 计算机系统结构与网络所, 陕西 西安 710049)

摘　要: 在主动网络安全原型的基础上,结合主动网络可动态加载协议的特点,提出了基于组件的主动网络安全机制。该机制能够适应主动网络保密性、完整性和可用性的安全需求,为主动节点和主动信包提供了可扩展加解密、信包验证、代码授权、代码执行监控以及代码撤销等一系列安全措施;同时支持动态加载和用户定制,使得主动网络可以根据具体应用设定不同的安全级别。

关键词: 主动网络; 组件; 可扩展加解密; 策略; 代码撤销
中图法分类号: TP393.08　　文献标识码: A　　文章编号: 1001-3695(2005)07-0111-04

Research of Component-based Security Mechanism for Active Networks

ZHOU Heng-lin, LI Zeng-zhi, WU Ya-qiang, LIAO Zhi-gang
(Institute of Computer Architecture & Network, Xi'an Jiaotong University, School of Telecommunication, Xi'an Shanxi 710049, China)

Abstract: Utilizing the trait that protocol can be deployed dynamically in active net, the component-based security mechanism is designed for active networks on the base of the security prototype of active networks. The mechanism can meet the security requirement of the active networks, such as secrecy, integrality and usability, and provides security measure including extensible encryption and decryption, authentication and authorization, execution monitor and code revocation, etc for active packet and active node. In addition, it supports dynamitic load and customization, therefore security level can be customized for active networks according to the actual application.

Key words: Active Network; Component; Extensible Encryption and Decryption; Policy; Code Revocation

主动网络是一种新型的网络体系结构,它提供一个软件的框架模型,使得网络中的应用可以由用户自己定制,简化了协议和服务的部署,同时也为网络管理、服务质量控制、可靠组播等提供了一条新的途径。然而主动网络的灵活性也使得主动网络的安全性面临更大的威胁。显然,如果安全性得不到较好的解决,该网络就不存在真正的实用价值。

本文在分析了主动网络特殊的安全需求的基础上,结合主动网络可动态加载协议的特点,提出了基于组件的主动网络安全机制,旨在为主动网络提供灵活、全面的安全服务。

1　基于组件的主动网络安全需求

1.1　主动网络的安全需求

主动网络安全需求主要有:

(1) 保密性。防止主动信包中的敏感内容被其他节点窃听。

(2) 完整性。防止主动信包的内容被非授权节点窜改。

(3) 可用性。防止恶意主动代码通过重新配置、修改、从内存擦除来破坏或改变节点的资源和服务;或者大量的主动代码不停地消耗网络连接或使用大份额的可用 CPU 周期,使得节点资源或服务过载。

收稿日期: 2004-06-22; 修返日期: 2004-09-02
基金项目: 国家自然科学基金资助项目(60173059)

1.2　基于组件的主动网络安全机制概述

从系统的角度分析主动网络面临的安全问题,可以抽象出主动网络的两个被保护对象,即主动节点和主动信包。为此,我们在主动网络安全原型的基础上设计了一套基于组件的主动网络安全机制。该安全机制包括四个安全组件:可扩展加解密组件、验证组件、授权组件以及代码撤销组件。其中,可扩展加解密组件为主动信包提供加解密服务;验证组件用来验证主动信包的身份和完整性;授权模块审核主动实体(包括 EE, AA 等主动代码以及主动信包)在本节点所拥有的权限,保护节点资源不被外来主动实体破坏;代码撤销模块则监视主动代码在主动节点中的执行情况,随时撤销非安全的主动代码。

基于组件的主动网络安全机制作为一种可动态加载的安全机制,它具有如下优点:

(1) 模块清晰。所有功能都被封装在一个单独的组件中,只通过端口与外界连接。应用不需要考虑组件内部代码的实现情况,只需要简单地设置与端口的连接就可以实现相应的安全功能。

(2) 可动态定制安全级别。为了适应不同的网络安全需求,应用可以通过组合不同的安全组件定制网络的安全级别。如网络性能比较差,可以只配置一个关键的安全组件;如果网络条件许可,也可以同时配置四个安全组件。

(3) 便于新的技术的加入。如果有新的主动网络安全技术出现,可以通过简单的改动某个相应组件或增加新的安全组件来升级主动网络的安全性能。

(4) 可以动态部署。主动节点可以不要事先部署好所有

的安全组件,只在需要的时候才从相应的节点获得所需的安全组件。这样,主动节点就可以定制与其预期通信的主动节点的安全级别。

基于组件的主动网络安全机制结构如图 1 所示。

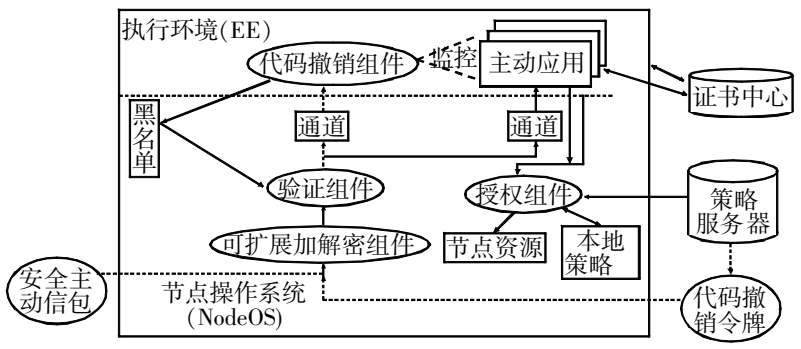

图 1 基于组件的主动网络安全机制结构图

(1) 安全主动信包:在普通主动信包基础上加入相应安全信息而构成的信包。

(2) 代码撤销令牌:为了实现代码撤销功能而设计的主动信包。

(3) 策略服务器:负责为主动实体生成相应的权限,并管理这些权限。

(4) 证书中心:认证各主动实体的身份,为其颁发证书和私钥。

(5) 本地策略库:记录外来主动实体在本节点所拥有的权限,该策略库是可以动态更改的。这是一个 RDBMS 形式的策略库。

(6) 黑名单:存放非安全主动代码的标志。

1.3 安全主动信包设计

安全信包的设计是在主动网络封装协议 ANEP 基础上增加了安全关联信息,其格式如图 2 所示。

(1) 证书链:存储一个证书序列,包括该主动信包的身份证书以及上一转发节点的证书。

(2) 签名链:存储一个签名序列,包括主动信包的签名和上一转发节点的数字签名。其中前者是针对信包内容(有效负载)的,表明信包身份;后者则是针对整个信包的,这是出于对相邻节点间完整性的考虑,保证信包内容在节点间不被非法窜改。

(3) 解码信息:会话密钥和所选择的加密方法用目的节点公钥加密后得到的密文。

1.4 策略服务器的设计

策略服务器需要具备两个功能:①考核主动实体的权限申请,并授予其相应的权限,同时服务器将备份所有主动实体的权限供主动节点查询;②策略服务器需要对权限进行管理,根据实际情况删除添加或者更改主动实体的权限,甚至撤销一段主动代码。

如果大量主动节点同时要求策略服务器提供服务,势必引起网络拥塞,甚至导致策略服务器瘫痪。为此,我们提出将策略服务器进行分层。其基本思想是:将策略服务器类似域名服务器(DNS)进行分层组织,如图 3 所示。每个主动网络域设置一个域内策略服务器,该服务器保存着该域内所有主动实体的权限信息以及最新被查询的一些权限;每个主动节点都与一个域内策略服务器相关联(为了保障可靠性,每个主动节点可另配置一个备用的策略服务器)。上层策略服务器存放所有与其相关的下层策略服务器的地址和权限映射信息。一个完整的权限查询工作过程类似 DNS 的域名查询。

版本号	安全模式	类型标志
主动信包头长度		主动信包包长度
源地址	目的地址	前一节点地址
选项		
有效负载		
证书链		签名链
解码信息		

图 2 安全主动信包格式

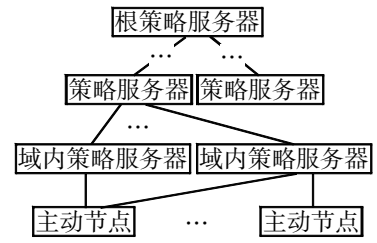

图 3 策略服务器层次图

为了实现代码撤销的功能,每个策略服务器上还保存一个撤销代码库,策略服务器可以通过汇报广播的机制来保持彼此待撤销代码库的一致性。所谓汇报广播的机制就是:下层策略服务器定期向自己的上层策略服务器汇报新的待撤销代码情况;上层策略服务器定期向自己的下层策略服务器广播新的待撤销代码情况。

2 主动网络安全组件关键技术研究

2.1 可扩展加解密组件

可扩展加解密指的是用户可以根据具体的网络状况和安全需求来选择加密算法。当网络性能比较差,对安全性的要求不高,那么可以采用比较简单的加密算法;如果当网络状态比较好,同时对安全性的要求比较高,可以采用 DESede 等比较复杂的加密方法。

主动网络的加解密与普通网络不同,普通网络可以对整个信包加密,而主动网络只对信包的敏感信息进行加密。敏感信息通常是指一些比较重要的只希望被目的节点获得的数据。至于主动信包中的主动代码是不需要加密的,否则主动代码无法在中间节点执行,这样就丧失了主动网络的灵活性。

一个加解密过程的流程如图 4 所示。

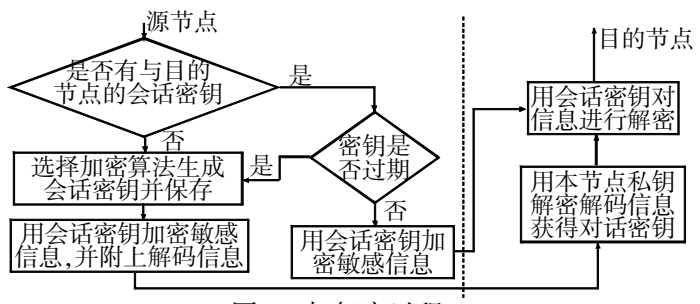

图 4 加解密过程

考虑到安全性,每个会话密钥都有一个生存周期,如果超过生存周期,源节点重新为本次会话生成会话密钥。

2.2 验证组件

验证组件要进行以下工作:

(1) 认证主动信包身份。主动信包证书、签名的有效性认证。

(2) 验证主动信包完整性。认证上一转发节点证书、签名有效性并检查此证书的拥有节点是否是信包前一节点字段所指的节点。

(3) 验证信包内主动代码的有效性。如果主动信包包含主动代码则还应查看该主动代码标志是否在黑名单中。

通过验证的主动信包才能被送往上层协议,否则被抛弃。

2.3 授权组件

授权组件是保障主动节点安全的关键组件,它决定了主动

实体可以访问的资源(如链路带宽、内存、CPU 时间以及通道等) 和进行的操作。在主动网络中, 每个 EE 在主动节点上执行, 需要访问主动节点的资源; 同时, 每个 EE 也拥有自己的资源和服务, AA 在 EE 中执行, 一方面要访问主动节点的资源, 一方面要访问 EE 的资源或引用 EE 提供的服务; 另外, 每个 AA 与主动信包是紧密相关的, 主动信包可能启动、挂起、调用或中止一段 AA 代码。可见, 主动网络中的被授权对象其实有三类: EE、AA 以及主动信包。

2.3.1　动态的访问控制机制

访问控制机制作用类似于一个安全沙箱, 任何主动实体的活动都被控制在安全沙箱的范围之内, 这样就不会对主动节点造成危害性后果。Java 提供的策略描述机制和 JAAS 可以灵活地实现主动网络的访问控制机制。

JAAS 可以用来实现主动实体的授权。它的工作原理如下:

(1) 主动实体在使用下层实体提供的 API 之前要先登录到下层实体, 出示自己的实体名, 这样下层实体可以获得当前登录的主动实体信息。如 AA 在使用 EE 提供的 API 之前要首先登录到 EE, 使得 EE 知道当前运行的是哪一个 AA 对象。在这里的实体名是可以从主动实体证书中提取。

(2) 当主动实体调用下层实体为其提供的 API 时, 必须以 doAS(Subject, Code) 或 doAsPrivileged(Subject, Code) 方法的形式调用。这里的 Subject 表示运行 API 主动实体对象, Code 代表此对象需要执行的代码。

(3) 主动实体执行代码时需要获得相应的权限。授权组件将查看本地策略库以考核此主动实体是否有执行此段代码并访问相应资源的权限。假如本地策略库中没有关于该主动实体的权限信息, 授权组件将向域内策略服务器请求该主动实体的权限。如果都查不到相关权限信息, 那么主动实体要么不允许执行此段代码, 或虽然允许执行, 但只能访问默认的最小资源。

传统的 Java 策略描述机制采用静态的策略文件, 无法动态更改, 也不能保证其安全性。为了能更好地适应主动网络环境, 我们必须对之加以改造和扩充。新的策略描述机制具有以下特点: ①ACL 采用三元组的标志形式为 < 主体, 组, 权限 >。其中, 主体即主动实体。组指明了主动实体所属的组别, 这里包括三类: EE 组、AA 组以及主动信包组。权限通常是一个 < 对象、行为 > 的二元组, 这里对象指的是主动实体操作的对象或访问的资源; 行为指的是主动实体对对象进行的操作。②对传统的 Java 权限集进行了扩充, 增加了一些新的权限类型, 如对主动节点路由表操作的权限, EE 内并发多线程的权限, 主动信包对 AA 代码操作的权限等。③采用 RDBMS 形式的策略库, 库中的权限信息是可以动态更改的。通过简单地编写 SQL 语句, 就能从策略库里查询或增删改权限信息。除了从策略服务器获得主动实体的权限, 节点用户也可以操作策略库中的权限, 当然, 只有管理员级别的用户才有这个权利。④授权信息存储在 RDBMS 中, 保障了授权信息的安全性。可以防止非授权用户操作本地策略库的授权信息。

2.3.2　安全限制技术

由于主动网络的灵活开放性, 导致主体与对象的关系变得十分复杂而且是动态变化的, 单纯地访问控制机制可能不能满足主动节点的安全需求。为此授权组件还向用户提供了多种安全限制技术, 使得用户可以动态定制自己的加强的安全策略。这里的限制技术包括时间限制 (如规定不同主动代码执行时间的上限), 范围限制 (如规定主动信包经过的节点总数超过某一值后只能访问默认的最小资源), 还有拷贝限制 (如规定每个主动信包拷贝自己的最高次数) 等, 这些在阻止主动信包独占节点资源方面是很必要的。

2.4　代码撤销组件

代码撤销组件运行在 EE 层, 负责监控主动代码的执行并撤销存在过失或安全隐患的主动代码。

2.4.1　监控技术

利用 DevPartner Java 版的测量工具(包括代码覆盖率测试 TrueCoverage、时间测试 TrueTime、内存测试 Memory Profiler) 可以对节点的内存占有率及 CPU 占有率进行实时测试。一旦发现其参数超过规定阈值, 即不良代码消耗资源过多时, 触发安全预警信息。释放该代码所占资源, 并将代码标志放入黑名单, 设置未处理标志。

2.4.2　协作式代码撤销(图 5)

由于主动代码的移动性, 已经发布到整个网络上的主动代码对于其创建者来说可能是不可知的, 代码撤销的关键在于主动代码的定位。采用全网广播方式即向整个主动网络公布被撤销的主动代码信息无疑是最快捷的方式, 然而缺点是容易产生数据风暴。这里我们提出了一种协作式的代码撤销机制, 能够比较方便地撤销主动代码, 同时不会给网络造成太大的负担。协作式代码撤销的基本原理是: 域内策略服务器发出代码撤销令牌在整个域内遍历; 接收到令牌的主动节点与令牌交互信息, 进行代码撤销。

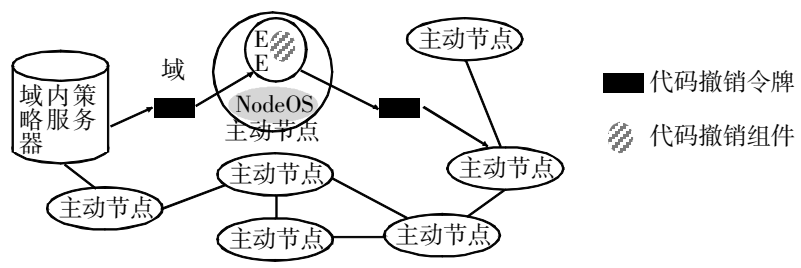

图 5　协作式代码撤销

代码撤销令牌是一种特殊的安全主动信包, 它包括以下几个关键字段:

(1) 待撤销代码。域内策略服务器向此字段填入需要被撤销代码的标志。

(2) 请求撤销代码。主动节点用此字段报告自己发现的需要被撤销的主动代码。

(3) 生存期。定义代码撤销令牌的生存期, 超过生存期的令牌将被抛弃。为了防止令牌丢失, 域内服务器如果在一个生存期内没有收到返回的令牌则重新颁发一个令牌。

我们从域内策略服务器和主动节点两部分来简要阐述协作式代码撤销的工作过程。

域内策略服务器的主要工作有:

(1) 发出新的代码撤销令牌, 在待撤销代码字段附上撤销代码库中未处理的代码标志。

(2) 对于接收到的令牌, 如果该令牌已经完成遍历, 则转 (3) , 否则向下一节点转发令牌。

(3) 将令牌中待撤销的代码标志在撤销代码库中设置为

已处理。

(4) 考核令牌请求撤销代码字段中的请求,通过考核的代码标志被放入撤销代码库,设置未处理标志。返回(1)。

主动节点进行代码撤销的过程如图 6 所示。

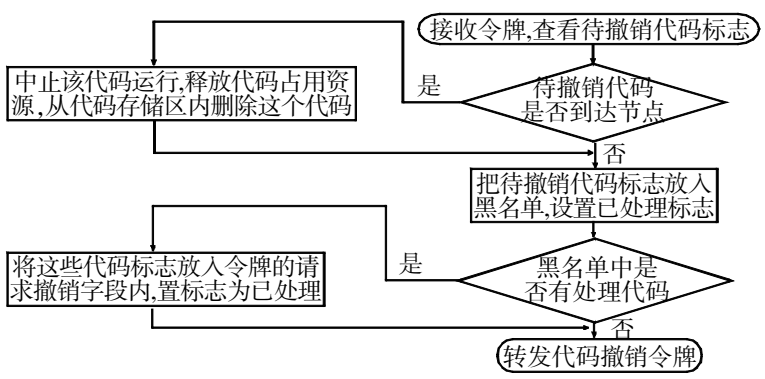

图 6 主动节点代码撤销过程

3 结论

基于组件主动网络安全机制在设计的过程中,考虑到了主动网络可能遭受的各种安全威胁,在此基础之上提出了相应的保护措施,为主动网络提供了一种比较全面的安全框架。更为重要的是,设计过程中吸收了软件设计模式的思想,保证了各安全组件可以被动态地进行配置,这样一方面提高了系统应用的灵活性,另一方面当有更好的安全技术出现时,可以很容易地被嵌入到系统中。

参考文献:

[1] ctive Network Working Group. Architecture Framework for Active Networks version 1 [R]. USA: Active Network Working Group, 1999. 17-38.

[2] Active Network Working Group. Security Architecture for Active Nets [R]. USA: Active Network Working Group, 2001. 13-54.

[3] Zhaoyu Liu, Roy H Campbell, M Dennis Mickunas. Active Security Support for Active Networks [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 2003, 33(4): 432-445.

[4] 寇亚楠,李增智,等. 主动网络安全原型的设计[J]. 电子学报, 2003, 31(11): 1702-1704.

[5] [美] Jamie Jaworski. Java 安全手册 [M]. 北京: 电子工业出版社, 2001. 218-242.

[6] 王建国. 主动网络关键技术研究[D]. 西安: 西安交通大学, 2002. 36-71.

作者简介:

周恒琳(1981-),女,江西赣州人,硕士研究生,主要研究方向为网络安全、网络管理;李增智(1938-),男,陕西人,教授,博士生导师,主要研究方向为网络管理及应用、分布式系统、电子数据交换;武亚强(1980-),男,陕西渭南人,硕士研究生,主要研究方向为网络管理、网络安全;廖志刚(1975-),男,陕西咸阳人,博士研究生,主要研究方向为主动网络安全。

(上接第 77 页)

(6) 将 LSP_{KH}, LSP_{HB}, LSP_{BD} 和 LSP_{DF} 连接起来形成 K 到 F 的 LSP_{KF}, 即 LER K-LSR3-LSR1-LER H-LSR8-LSR7-LSR6-LSR4-LER B-LSR10-LSR11-LSR12-LER D-LER F。这时 H, B, D 这些边缘 LSR 除了维护 MPLS 域的 LSP 之外,还要负责数据的转发和标签的转换,由于它们不是 LSP_{KF} 的出口 LER,因此要继续进行转发。

4 多层次 MPLS 流量工程网络的几点说明

(1) 类似于边缘 LSR B, LSR D, LSR F, LSR H 等在 LSP 建立过程,不仅仅是某一层的某一个 MPLS 域的边缘 LSR,而且是多层多个 MPLS 域的边缘 LSR,如 B 既是第一层 MPLS 域的边缘 LSR,也是第二层 MPLS 域(1)的边缘路由器。

(2) B 节点不仅负责和 H 节点的数据交换,而且还要和 E, F 节点的数据交换,因此 B 节点容易造成瓶颈。因此应该增加 B 节点的交换速度,或者增加更多的边缘 LSR 和 B 同时工作,分担 B 的压力。

(3) LSP 不再是某一个域中的 LSP,而是每个域中的根据 CR-LDP 或者 RSVP 这些协议确定的 LSP 的连接。

(4) 有的边缘 LSR,如 LSR B, LSR D, LSR H 在确定自己的 LSP 的同时,还要负责数据的转发,也就是说,数据到达这些边缘 LSR 的时候,标签不再被剥去,而要换上标签,向另一个 MPLS 域中转发。当然,如果将所有 MPLS 域中 LSP 连接起来之后,将所有 MPLS 域中的明显路由都交给边缘 LSR K 集中控制,那么 LSR B, LSR D, LSR H 就和核心 LSR 功能相同。

(5) 边缘 LSR 要能够根据目的地址确定建立域内 LSP 的出口 LSR。

5 结束语

由于 MPLS 以集成模型方式具备了重叠模型的全部功能,并且为流量工程的自动实现提供了可能性,因此,在各种实现流量工程的方案中,MPLS 无疑是当前比较好的解决方案。但是为了 MPLS 应用范围更广,特别是目前网络速度发展很快的情况下,尽可能让 MPLS 在更大的范围里面使用,尤其是信息化密集的地方,在这种情况下,多层次的 MPLS 流量工程的网络无疑成为一种更好的方案。

当然,这种方案还有很多地方需要进一步完善,主要是边缘 LSR 的改进,要求边缘 LSR 能够完成角色的转变,同时,随着用户数量的增加,那么如何对这种多层次 MPLS 进行扩展。

参考文献:

[1] ipeng Xiao, Alan Hannan, Brook Bailey. Traffic Engineering with MPLS in the Internet[J]. IEEE Network, 2000, (3/4).

[2] Awduche D, *et al*. Requirements for Traffic Engineering over MPLS [S]. RFC 2702, 1999.

[3] 李小东. MPLS 技术与实现[M]. 北京: 电子工业出版社, 2002. 31-47.

[4] 谭咏茂,等. DiffServ over MPLS 模型中的流量工程机制[J]. 武汉大学学报(理学版), 2002, 48(1): 47-50.

[5] 张艳,郑纪蛟. 基于 MPLS 的流量工程[J]. 计算机应用研究, 2002, 19(2): 58-60.

作者简介:

许先斌(1954-),男,湖北枝江人,教授,博士,主要研究方向为计算机网络与通信、分布并行处理及海量信息存储;袁行船(1979-),男,硕士研究生,主要研究方向为计算机网络与通信。