计算机科学 2001Vol. 28№. 8

基于安全数据结构的防火墙

Firewall Based on Security Data Architecture

原 箐 卿斯汉

(中国科学院信息安全工程技术研究中心 北京 100080)

Abstract With the improvement of network, security becomes more importable. In market, firewall is one of the importable security products. Firewall can prevent your network from being attacked, for example IP snoop, mail-spm etc. But firewall can not prevent itself from the action which attacks the firewall's code and firewall system. So we add integrity services into firewall system. In firewall system, every module does self-check integrity before it is loaded by system. A bilateral authentication procedure is designed for two modules which call each other to establish trust in the identity and integrity of each other. And it is still necessary to ensure secure linkage between the two parties. Firewall system establishes on the security data architecutre.

Keywords Firewall, Self-check integrity, Bilateral authentication, Secure linkage, Security data architecture

一、引言

防火墙技术是一种网络安全技术。它是在受保护网与外部网之间构造一个保护层,把攻击者挡在受保护网的外面。这种技术强制所有出入内外网的数据流都必须经过此安全系统。它通过监测,限制或更改跨越防火墙的数据流,尽可能地对外部网络屏蔽有关受保护网络的信息和结构来实现对网络的安全保护。但是如果一个黑客利用防火墙本身操作系统的漏洞来对此防火墙进行破坏,例如:修改、替代或删除防火墙的模块、规则配置文件或认证用户库,那么这样的防火墙就没有安全可言了。所以为了保证防火墙本身的安全性,我们在防火墙系统里加入了自身完整性检测、双向认证、安全连接等机制,使防火墙系统建立在一个安全、可信赖的架构上(我们把这种架构叫做安全数据结构)。另外,这种安全数据结构还为防火墙系统提供了一种灵活的加载模块机制。

二、防火墙结构及原理

防火墙是一种网络安全技术,最初它被定义为一个实施某些安全策略保护一个可信网络,用以防止来自一个不可信的网络的攻击的系统或系统组。

防火墙的工作原理是按照事先规定好的配置和规则,监控所有通过防火墙的数据流,只允许授权的数据通过,同时记录有关的联接来源、服务器提供的通信量以及试图闯入者的任何企图,以方便管理员的监测和跟踪,并且防火墙本身也必须能够免于渗透。

防火墙一般分为包过滤和代理服务器两个类型。包过滤式的防火墙是在IP层,它会检查所有通过信息包里的IP地址、端口号及其它包头信息,并按照系统管理员所给定的过滤规则过滤信息包。代理服务器是在TCP层,对TCP层上的服务进行代理,如FTP、HTTP、TELNET、SSL等等。这样防火墙可以提供更多的控制,例如可以对用户、对每个TCP服务的命令进行控制。

三、安全数据结构的原理

3.1 安全数据结构原理

安全数据结构是一种逻辑结构,为一套软件或系统提供了一种安全模式。具体地讲,安全数据结构规定了一系列规则,系统在完成它的功能时必须遵循安全数据结构的规则,来保证该系统本身的安全性。到底安全数据结构提供了一套什么样的安全模式?首先,系统在运行前,它的各个模块都要进行初始化,即每个模块都要进行数字签名以便以后进行完整性检测。然后,模块之间调用前都要进行自身完整性检测和双向认证,以保证模块没有被修改过。并且在模块调进内存后模块还要进行一次自身完整性检测,保证在装载过程中没有被修改。模块在内存中要随时进行安全连接检测,保证没有在内存中被修改。另外,在调用函数时还可以进行安全连接检测(考虑到速度问题,该检测方法可不选)。

3.2 自身完整性检测原理

完整性检测是用来检测对象的完整性，即该对象是否被修改过，完整性检测的具体过程大概分为三步：

(1)先由系统为该对象生成一密钥对。

(2)用私钥对该对象进行签名：先用一个指定的hasb 算法对该对象进行 hash，得到该对象的一个摘要；然后，用私钥对这个摘要进行加密；最后，保存产生的密文。

(3)对该对象进行完整性检测：先用公钥对该签名进行解密得到摘要 obj1；同时用指定的 hash 算法对该对象进行 hash 得到 obj2；把 obj1 与 obj2 进行比较，如果相等则完整性检测成功；反之，失败，说明该对象已被恶意地修改。

自身完整性检测，是对象自己来进行完整性检测。该对象的数字证书是在它一加入到系统里时，就由系统产生了。并且也是由系统来对它进行签名的。系统把该对象的清单保存起来，清单包括该对象的签名、hash 算法、证书和该对象在硬盘(或内存)上的物理地址。

当模块被上载到内存前，或被调用前都要进行一次自身完整性检测，以保证该模块是可信赖的。首先，该模块从系统中取到自己的清单，从清单中得到自己的公钥、hasb 算法以及该模块的签名。利用这些信息模块对自己进行完整性检测。

自身完整性检测是个程序，它存在各个模块中。

3.3 双向认证原理

双向认证是在模块之间调用时进行的，是用来证明调用者与被调用者都是可信赖的。双向认证的过程如图 1 所示。

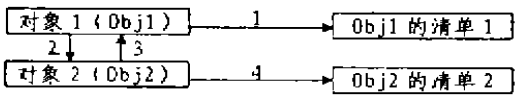

图 1

假如对象 1 要调用对象 2，则对象 1 要与对象 2 进行双向认证，一般分四步：第一步，对象 1 进行自身完整性检测；第二步，对象 1 验证对象 2；第三步，对象 2 进行自身完整性检测；第四步，对象 2 验证对象 1。

当上面四步都成功后，双向认证就城功地完成了，对象 1 与对象 2 认为对方是安全的。

3.4 安全连接原理

当对象被系统调到内存中后，调用该对象的功能时要检测调用指针是否在该对象的内存地址范围内。该检测是可以配置的。

四、基本安全数据结构的防火墙结构和原理

4.1 基于安全数据结构的防火墙的结构

为了解决防火墙的自身安全问题我们把防火墙建立在安全数据结构上。我们把防火墙分成多种模块，每个模块的建立、调用都要遵循安全数据结构的机制。基于安全数据结构的防火墙结构图如图 2 所示。

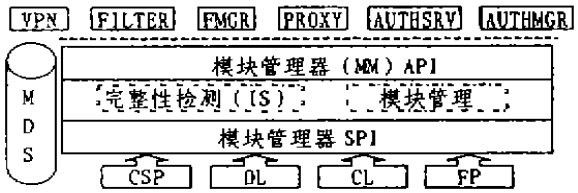

图 2 基于安全数据结构的防火墙结构图

图 2 中，MDS：模块目录服务--用来记录各个模块的清单以及其他一些信息。MM：模块管理器--是该防火墙的核心部分，用来对该安全数据结构进行管理。CSP：密码服务提供者--用来对上层服务提供加密、解密、产生密钥对等操作。CL：证书库--用来保存各个模块的证书。DL：数据存储库。提供数据存取服务。FP：防火墙规则--提供防火墙配置规则。PROXY：防火墙代理服务器--包括：FTP 代理、HTTP 代理、TELNET 代理等。FILTER：防火墙包过滤--用来提供 IP 层的过滤。VPN：虚拟私有网。FMGR：防火墙的规则配置管理器。AUTHSRV：防火墙的用户认证服务器。AUTHMGR：防火墙的用户认证管理器。SPI：服务提供接口。API：应用提供接口。IS：完整性检测服务器。用来进行完整性检测。

从该防火墙的结构图中，可以看到我们把该防火墙分成三个基本层。①系统安全服务层。它为系统提供一些安全服务；②模块管理器；③应用服务提供层。它是上层服务，为客户端提供各种服务。

4.2 基于安全数据结构的防火墙的主要模块

4.2.1 *系统安全服务层* 该层的模块是用来为上层应用提供一些基本服务的。本系统把基本服务分成四种：

- 密码服务(CSP)。它为上层应用提供的服务有：产生密钥对、加密/解密、签名/认证、摘要/hash、产生随机数、用户认证算法(Skey、securid、enigma 等)。
- 数据存储库(DL)。它为所有需要数据存取的模块提供服务。它提供了一般的存储管理机制，有创建、打开、删除、关闭操作；提供了 ACL(访问控制链)基础的操作，有创建、打开、建立关系等；还提供了一般操作，有插入、删除、修改、查找等。
- 证书库(CL)。支持一些常用的证书格式，如 X.509、PGP、SET 等。基本操作有为本系统产生证书、签名、验证；从证书或证书链中得到有用的部分；缓存部分被处理过的证书。

·防火墙规则(FP)。为上层的防火墙代理服务模块、包过滤模块、VPN 模块提供配置规则,用来控制防火墙的行为。它是由上层防火墙规则管理器(FMGR)来进行管理的。

这四种基本服务在提供自己的服务时,可能要用到其它服务,如 CL 提供服务时要用到 CSP 的一些服务。它们之间是可以互相调用的,但必须经过 MM 代理,如图 3 所示。

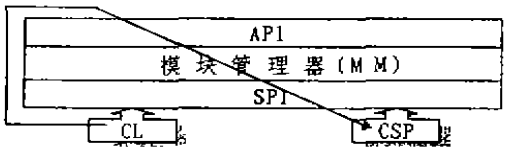

图 3 CL 调用 CSP 提供的服务过程

4.2.2 应用服务提供层 应用服务提供层是用来提供防火墙服务的,它包括:应用层代理服务器(FTP、TELNET、HTTP、SMTP、NNTP、FINGER 等)、包过滤(FILTER)、VPN、NAT、防火墙规则管理器、防火墙认证服务器、防火墙用户管理器。当客户端向应用(APP1)提出服务请求时,APP1 向客户提供服务,如果服务需要底层服务提供时(如:代理服务器要到 FP 中查找配置规则)、APP1 通过 API 接口向 MM 提出请求,MM 再通过 SPI 接口向 FP 提出请求,FP 再通过 MM 向 APP1 提供服务。

4.2.3 模块管理器(MM) 模块管理器是该系统的核心,它为所有的模块提供安全服务,即为应用服务提供层和系统安全服务层提供动态的管理,提供一般的完整性服务。

模块管理器的 API 接口可以逻辑地被分成下面三个功能:核心服务;上下文管理服务;基本模块管理。

核心服务包括发现和连接系统安全服务模块。上下文管理服务是允许一个应用为一个系统安全服务模块维持一个算法设置列表。基本模块管理提供把服务分配给下面四种基本服务的机制。

另外,MM 提供一系列的完整性服务,这些完整性服务是供 MM、模块管理器、系统安全服务模块(CPS,DL,CL,FP),上层应用使用的。它们用来对自身和其他模块进行完整性检测,MM 还提供了安全线程机制——单写多读。

下面详细地介绍 MM 的一般模块管理功能、基本模块管理功能和完整性服务。

1)MM 的一般模块管理功能。MM 一般模块管理由两部分组成:模块目录服务(MDS)和动态上载。

·模块目录服务(MDS)。MDS 存放着各个以被安装在本系统上的组件的清单以及其它一些信息。MDS 还管理一个对象路径,该路径记录着每个已安装的服务模块、应用、没上载的服务模块和 MM 的组件的信息。MM 可以通过该路径查找到各个模块的位置,MDS 服务对每个模块都是公开的。MDS 提供的完整性服务功能如图 4 所示。

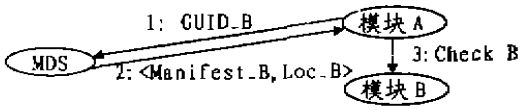

图 4

·动态上载(Dynamic Loading)。MM 用一系列的 API 接口来管理服务提供模块。主要的管理接口是 MM_ModuleLoad 和 MM_ModuleUnLoad。具体过程如下。应用程序(APP)需要下层的服务提供模块(SPI)的服务。首先,APP 通过 MM_ModuleLoad API 接口向 MM 提出请求,MM 与 APP 之间进行双向认证。成功,MM 通过一个 SPI 接口 MM_ModuleSPILoad 向服务提供模块 SPI 提出上载请求,在请求被答应之前 MM 与 SPI 之间进行双向认证。都成功后 SPI 被上载到 APP 的内存空间中,在这时还要对 SPI 进行一次自身完整性检测,用来防止在上载过程中 SPI 被恶意修改。在上面过程中任意一个检测出错都会使该次上载过程中断,并且记录发生错误的模块,防止其它模块调用它,同时向安全管理员发出警报。

2)MM 的基本模块管理。该防火墙系统定义了四种基本服务模块,MM 针对这四种基本服务提供了四种基本模块管理:密码服务模块管理;证书库模块管理;数据存储库模块管理;防火墙规则模块管理。

MM 在系统的整个生命期间都维持这些管理功能,并且把它们的 API 接口提供给应用。

3)完整性服务。在一个模块上载和初始化前,MM 都要用该模块的清单来验证一下该模块证书的可靠性以及该模块代码的完整性。双向认证是通过完整性检测,在两个模块之间建立可信赖关系。在该系统中各个模块的清单是在模块安装前产生的并保存在 MDS 中,进行完整性检测时从 MDS 中得到所需要的清单。

在模块上载到内存中前和应用程序调用服务时,都要与 MM 进行双向认证。MM 端的过程:

·MM 进行自身完整性检测。

·MM 对模块进行完整性认证(验证模块在硬盘上的映像,模块在内存中的签名)。

·MM 验证安全连接(判断模块的初始接入点是否在该验证模块的物理地址范围内)。

模块端的检测过程:

·模块自身完整性检测。

·从 MDS 中得到 MM 的清单。

·验证 MM 的完整性。

·判断为 MM 返回的地址是否在自己的范围内(进行安全连接检测)。

·MM 代理 callbacks。

下面举个例子来说明整个服务过程。

4.3 FTP 代理服务器的工作过程

在该防火墙系统中的代理服务器、认证服务器、认证用户管理器、CSP、DL、CL、FP、MM 等模块在加载到防火墙系统前,都要先到我们的 CA 申请一对密钥,用私钥和防火墙自己的 hash 函数对该模块进行签名,然后用 MST 对其产生清单,把该清单保存在 MDS 中。MDS 运用了 DL 提供的存取机制。防火墙系统建立后,要想在其上增加新的功能,那么这个新模块要先产生清单后,再要进行自身完整性检测,成功后该新模块才能加载到防火墙系统上。

下图简单说明了需要认证的 FTP 代理服务器的执行过程。

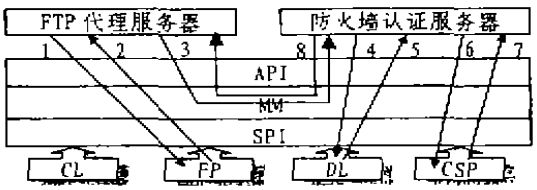

图 5 FTP 代理服务器提供服务的过程

1# 客户端向 FTP 代理服务器提出服务请求。FTP 代理服务器要检查自己的规则配置来决定如何处理该请求。FTP 代理服务器通过 MM_ModuleLoad (FPGuid)API 接口向 MM 提出请求,MM 就通过 MM_ModuleSPILoad(FPGuid)SPI 接口把 FP 模块上载到 FTP 代理服务器的内存中(在这过程中,FTP 代理服务器与 MM,MM 与 FP 之间都要进行自身完整性检测、双向认证和安全连接检测)。

2# FP 模块把 FTP 代理服务器的配置返回给 FTP 代理服务器。

3# FTP 代理服务器通过规则来决定是否给客户端提供服务。如果配置规则规定需要进行用户认证,那么 FTP 代理服务器通过 MM 向防火墙认证服务器提出认证请求(在过程中 FTP 代理服务器与 MM,MM 与防火墙认证服务器之间都要进行自身完整性检测、双向认证和安全连接检测)。完整性检测通过后,防火墙认证服务器向 FTP 代理服务器提供服务。

4# 防火墙认证服务器通过 MM_ModuleLoad (DLGuid)API 接口向 MM 提出请求,MM 通过 MM_ModuleSPILoad(DLGuid)把 DL 上载到内存中,在这过程中也要进行完整性检测。

5# 防火墙认证服务器从 DL 中取出用户信息。

6#7# 与 4#5# 相似,认证服务器根据用户信息从 CSP 中取出相应的认证算法。然后,认证服务器用 CSP 提供的认证算法对客户端的用户进行用户认证。

8# 防火墙认证服务器把结果通过 MM 返回给 FTP 代理服务器。FTP 代理服务器根据结果来决定对客户的操作。

结束语 基于安全数据结构的防火墙系统的提出是为了保证防火墙本身的安全性。为了提高防火墙的运行速度,其中的一些完整性检测是可要可不要的(例如:安全连接检测,还有一些模块你认为非常安全,它们之间的调用是安全的,那么它们之间的双向认证就可以省略),本系统使防火墙的各个模块之间的安全性得到延伸,并且具有很大的扩展性,新的算法、新的功能可以随意加载同时保证了它们的安全性。

参 考 文 献

1 王家业,荆继武,朱森存. 包过滤防火墙的安全研究. 计算机科学,1999,26

2 Begel A,et al BPF+:Exploiting Global Data-flow Optimization in a Generalized Packet Filter Architecutre. ACM SIGCOMM Preview,1998

3 Srinivasan V,Sun S,Varghese G. Packet Classification using Tuple Space Search. ACM SIGCOMM Preview, 1999

4 Srinivasan V,et al. Fast and Scalable Layer Four Switching. Proc. of Sigcomm,1998

5 Intel CDSA. Available at,http://www.opengroup.org/onlinepubs/009608599/index.htm

6 Nilsson S,Karlsson G. IP Address Lookup Using LC-Tries. Available at:http://www.nada.kth.se/~snilsson/public/papers.html

7 Chapman D B,Zwicky E D 著 舒若平,朱孝明,郑宏,胡红宇译. 构筑因特网防火墙. 电子工业出版社

8 Andersson A,Nilsson S. Faster searching in tries and quadtrees-analysis of level compression In:Proc. of the Second Annual European Symposium on Algorithms

9 鲁士文编著. 计算机网络原理与网络技术. 机械工业出版社

10 Bailey M L,et al. PATHFINDER:A pattern-based packet classifier. In:Proc. of the First Symposium on Operating Systems Design and Implementation,1994

11 Andersson A,Nilsson S. Implementing Radixsort. Available at:http://www.nada.kth.se/~snilsson/public/papers.html

12 Andersson A,Nilsson S. Improved Behaviour of Tries by Adaptive Branching. Available at:http://www.nada.kth.se/~snilsson/public/papers.html