第34卷 第20期 计 算 机 工 程 2008年10月
Vol.34 No.20 Computer Engineering October 2008
·网络与通信· 文章编号：1000—3428(2008)20—0109—03 文献标识码：A 中图分类号：TP393.02

基于 Linux 的 XFRM 框架下 IPSec VPN 的研究

阚 闯，栾 新，戚玮玮
(中国海洋大学信息科学与工程学院，青岛 266100)

摘 要：针对现有 IPSec VPN 系统在效率和可靠性方面存在的问题，提出并改进了一种基于Linux最新内核平台的 IPSec VPN网关系统。给出Linux 的XFRM 框架结构和函数调用结构的表述，其中包括XFRM框架模块与内核中IPSec进入外出处理的交互结合和VPN 网关安全隧道的构建，利用 XFRM 框架实现 IP 层处理和IPSec 处理。对新系统进行了仿真实现与性能评价，结果表明，它是可行和有效的。
关键词：虚拟专用网；IPSec 协议；XFRM 框架结构；PF_key 协议

Research on IPSec VPN Under Framework of XFRM Based on Linux

KAN Chuang, LUAN Xin, QI Wei-wei
(Institute of Information Science and Engineering, Ocean University of China, Qingdao 266100)

【Abstract】 On the basis of the efficiency and reliability feature of existing IPSec VPN systems, the architecture of an IPSec VPN gateway system on Linux newest kernel platform is presented and improved. The corresponding Linux XFRM structure and function call structural description are introduced, including XFRM framework interaction with IPSec module handling in the kernel and the VPN gateway security tunnel construction. This system uses Linux XFRM frame to combine IP process and IPSec process. Simulation results show that the new system is both feasible and effective.
【Key words】 Virtual Private Network(VPN); IPSec protocol; XFRM framework; PF_key protocol

XFRM框架下的IPSec VPN系统是在基于Linux 最新内核 2.6.19 平台[1]上实现的。相比于内核 2.4，IPSec处理模块已集成在内核主线中，IPSec是VPN最常用的第 3 层隧道安全协议，但目前的IPSec VPN产品在应用上仍存在许多有待克服的问题，利用Linux的Netfilter框架来实现基于IPSec的VPN系统，利用Netlink套接字实现用户空间与内核空间IPSec部分进行通信。IPSec作为一个楔子插入网络层和数据链路层之间，需要重复实现网络层的很多功能，效率低，不够流畅；不能够保证VPN的高效性和可靠性，传统的IPSec实施方案难以保证 VPN系统的通信性能和可扩展性。因此，本文提出利用 Linux 的XFRM网络框架来实现基于IPSec隧道模式下的VPN系统，利用PF_key协议结构的通信接口，与IPSec处理紧密结合，实现传输隧道上数据包的加密、认证和封装等。

1 IPSec VPN 相关技术

实现VPN的隧道协议有多种，其中的IPSec是一个协议族[2]，提供了一种标准、健壮且包容广泛的安全机制，既可以用来保护IP上层协议(如UDP和TCP)，也可以用来保护一个完整的IP数据包。这 2 方面的保护分别由IPSec的传输模式和隧道模式来提供。

认证报头(Authentication Header, AH)为 IP 数据包提供无连接的数据完整性和数据源身份认证，具有抗重播攻击的能力。封装安全载荷(Encapsulating Security Payload, ESP)为 IP 数据包提供加密、无连接的数据完整性、数据源身份认证以及抗重播攻击保护。AH 与 ESP 类似，但它不提供机密性服务。

IPSec 保护一个 IP 包之前，必须先建立一个安全关联(Security Association, SA)，SA 定义如何对一个特定的 IP 包进行处理。对一个外出包而言，根据 IP 头部信息可以找到安全策略库(Security Policy Database, SPD)中与之对应的一个或多个安全策略(Security Policy, SP)，而每个 SP 指向一个或多个 SA 或者 SA 束，安全关联数据库(Security Association Database, SAD)对不同的安全协议所对应的 SA 进行管理。

2 IPSec VPN 系统设计

2.1 IPSec VPN 系统总体结构

与传统的 IPSec VPN 安全网关相比，新的 IPSec VPN 实现原理框图主要由 4 部分组成，如图 1 所示。

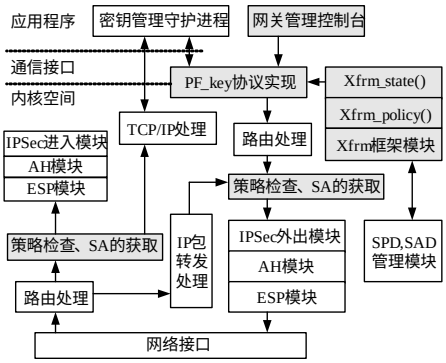

图 1 IPSec VPN 网关系统总体结构

在图 1 中，灰色显示的模块为 Linux 内核 2.6.19 平台下本系统新添加的 IPSec VPN 功能模块，图中的策略检查、SA 的获取模块实际都属于 XFRM 模块，调用的是 XFRM 所提供的函数；XFRM 网络框架为 IPSec 处理中策略的选择提供

作者简介：阚 闯(1978－)，男，硕士研究生，主研方向：智能嵌入式网络与信息安全；栾 新，教授、博士生导师；戚玮玮，硕士研究生
收稿日期：2007-11-30 **E-mail：**kansir2006@163.com

依据，实现了 IPSec 的 SPD/SAD 的管理，它又与原网络框架的路由和网络数据处理密切关联；IPSec 模块负责构建 VPN 安全隧道；网关管理控制台提供操作界面；而 PF_key 套接字接口模块实现用户进程与内核间的通信。下面对 TCP/IP 协议栈中涉及到 IPSec 的主要模块的实现情况进行分析。

2.2 密钥管理守护进程和网关管理控制台

应用层密钥管理守护进程使用 PF_key 套接口与系统内核中的密钥引擎进行通信，密钥引擎或 SAD 是一个逻辑实体，它对不同的安全协议所对应的 SA 进行管理；各种安全协议通过内核内部的逻辑接口请求并且获得对应的 SA。网关管理控制台主要提供操作界面，接收并分析操作命令，构造消息与 PF_key 接口交互，进而达到和内核交互的目的。

2.3 PF_key 协议通信接口模块

PF_key是一个新的套接口协议族[3]，用于可信赖的、有特权的密钥管理程序和操作系统内部密钥管理者之间的通信。对于Linux 2.6.19 内核PF_key协议族定义在/usr/include/linux/socket.h中，接口模块实现并扩展了PF_key协议族，接口模块主要完成 2 大功能：(1)接收用户进程的消息，处理内核的SAD, SPD，并给用户进程反馈一个消息；(2)IPSec处理过程中发现SA过期等情况时，利用PF_key提供的函数构造消息，并将消息写到用户守护进程的socket接收队列中。

2.4 XFRM 模块

Linux 2.6 版内核的一个重要改进就是引入了一种新的IP包处理的网络框架XFRM，XFRM代表传输(transformer)，定义在目录/usr/include/net/xfrm中，其框架结构如图 2 所示[4]。其中，定义 2 个结构xfrm_policy{}表示IPSec SP，xfrm_state{}表示IPSec SA；xfrm_state{}通过xfrm_templ{}和xfrm_policy{}关联；SPD由xfrm_policy{}结构链组成，SAD由xfrm_state{}结构链组成。

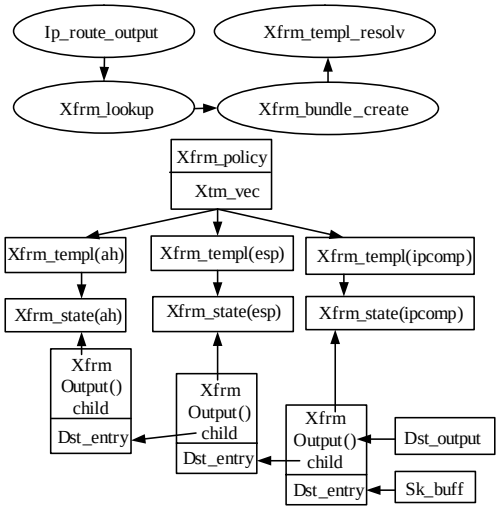

图 2 XFRM 框架结构和函数调用结构

3 IPSec VPN 系统实现

本系统实现的重点在于 XFRM 框架模块与内核中 IPSec 进入外出处理的交互结合和 VPN 网关安全隧道的构建。除了安全问题，效率和稳定性问题是系统实现所要考虑的关键问题之一，在以往的 IPSec 实现中，对于每一个经过 VPN 网关的数据包，都必须查询 SPD 和 SAD 来确定用于处理该数据包的 SA，这种查询开销似乎不可避免。而在本系统中，XFRM 模块与 IPSec 处理相结合节省了大量的 SA 获取的时间，提高了传输效率，传输过程中数据丢包率大大减少。

首先初始化SPD和SAD，SPD和SAD是IPSec处理的基础，需要定义并初始化相关的数据结构；注册PF_key协议。在内部网络的数据包发送到公网之前，本系统的IPSec模块对数据包进行加密、认证和封装；收到数据包时，用相应的策略和算法进行解包工作。这就构建了VPN安全隧道[5]。

IPSec 外出模块由外出处理模块与 XFRM 框架模块结合完成，处理过程如下：

(1)查询 SPD，得到对数据包处理的 3 种可能：丢弃，直接发送，应用 IPSec。

(2)若需要应用 IPSec，在路由缓存表中查找路由，若无则调用 ip_route_output 在 FIB 中查找路由，IPSec 的外出模块中会调用 xfrm_lookup()在 SPD 中查找 xfrm_policy{}，此时栈中的 dst{}指向原 dst{}结构。然后 xfrm_lookup()调用 xfrm_tmpl_resolve 从 xfrm_policy{}中解析得到 xfrm_tmpl{}结构，xfrm_tmpl{}中包含了数据包的处理方式，并查找与 xfrm_tmpl{}匹配的 xfrm_state{}，这个过程相当于查找与 IPSec 策略对应的 SA 或 SA 束。这样就创建了 sk_buffer 的 dst 束；Xfrm_bundle_create 创建堆叠式目标和 SA 束。当数据包建立后调用 dst_output()，调用 dst 束中的每个 dst 的 output 函数，循环处理直到遍历 dst 束中的每个 dst，此时 dst{}结构中外出例程函数指针指向的外出处理函数被调用，外出函数从 sk_buff{}的 dst{}中可得到 xfrm_state{}。最后由原 dst 的外出函数 ip_output 发送数据包，此时的数据包经过了 IPSec 的处理已封装成另一 IP 数据包了，其调用过程可参照图 2。

(3)循环对外出包进行 IPSec 头的填充，数据的加密及数据认证。

IPSec 进入模块：XFRM 框架结构的输入部分比输出部分简单，XFRM 输入函数的处理和上层协议(TCP, UDP)的处理一样。主要流程为：(1)按 IP 头的次序处理 IP 包；(2)处理已建立的 IPSec 部分；(3)检查 IPSec SP；(4)将 IP 包转到下一个处理。

4 IPSec VPN 网关原型系统测试与分析

本系统在 Linux 平台下模拟实施了 IPSec 策略的安全网关，一端与受保护的内网相连，另一端与不安全的外网相连。2 个跨越外网的安全网关 GWA(外网：eth1:192.168.1.1 内网：eth0:192.168.2.1)，GWB(外网：eth1:192.168.1.2 内网：eth0:192.168.3.1)就构建出 VPN 环境，2 个子网内主机 HostA (192.168.2.0), HostB(192.168.3.0)分别通过 AH 隧道模式、ESP 隧道模式建立 VPN，为子网中的终端提供数据认证及加密，网关均用安装双网卡的 PC 机模拟，每块网卡连接一个网段。其拓扑图如图 3 所示，将 VPN 配置成路由器，以具有数据包的寻路、转发功能。

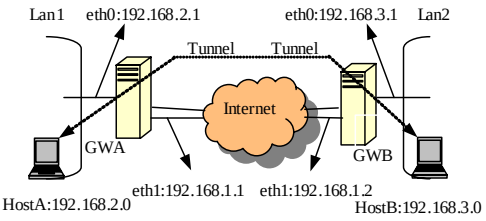

图 3 Linux 网关间 IPSec 隧道模式下 VPN 拓扑图

程序调试成功之后，截获了相应网段中的 IP 信包，并绘出了 AH, ESP “隧道传输时间”的统计数据，如表 1 所示。在 2 个内网之间传输文件，分组测试数据(每组 5 个数据)，最后得到 7 组平均值。

表 1　使用 AH, ESP 隧道传输时间统计数据

Size/MB	$T_{no\text{-}ipsec}$/s	T_{ah}/s	T_{esp}/s
2.802	2.987	3.825	4.062
4.866	5.068	6.356	6.750
6.350	6.853	8.137	8.659
8.858	9.503	11.225	12.081
10.740	11.416	13.359	14.300
15.973	16.934	19.837	20.956
24.001	25.353	29.515	31.578

假设不使用IPSec安全隧道传输的时间为$T_{no\text{-}ipsec}$，使用AH隧道传输时间T_{ah}，使用ESP隧道传输时间T_{esp}，最终绘出传输时间统计结果曲线的对比图，如图 4、图 5 所示。

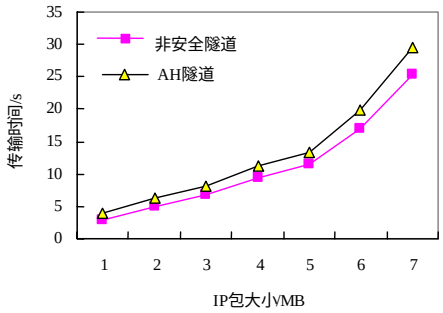

图 4　T_{ah}与$T_{no\text{-}ipsec}$对比图

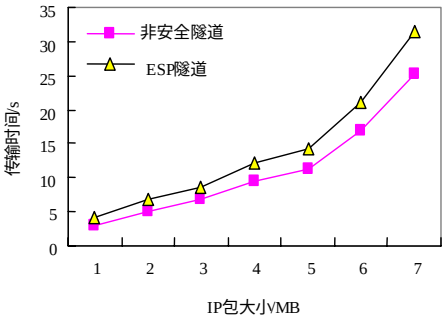

图 5　T_{esp}与$T_{no\text{-}ipsec}$对比图

另外，为了测试该 VPN 网关在长时间、高负荷情况下的稳定性，在进行文件传输时可以进行 ping 操作，例如在 HostA 上进入 DOS 环境运行命令“ ping 192.168.3.0 -t ”，结果如下：

Ping statistics for 192.168.3.0:
Packets: Sent = 6350, Received = 444.5, Lost = 590.5 (7% loss),
Approximate round trip times in milli-seconds:
Minimum=8137ms, Maximum=8659ms, Average=8558 ms

结果表明数据传输过程中有 7%的丢包现象发生但不是很严重，可以正常工作，稳定性基本符合要求。

可以得出结论，在使用 VPN 的情况下，文件读取速度慢于不使用 VPN 的情况，约为不使用 VPN 速度的 73.43%，网关之间的隧道已建立成功，即 VPN 已形成，网关所在的 2 个子网内的终端可以进行安全的数据传输。

5　结束语

本文通过对 Linux 2.6.19 内核的重新编译，实现了一种新的网络框架 XFRM 下的 IPSec VPN，源代码文件中增加了 net/xfrm 目录，包括算法、进入、外出、SP 和 SA 的处理代码，并在 net/ipv4 中有 XFRM 相关的处理文件、包含 AH 和 ESP 的相应处理的代码，有关路由处理的代码也作了相应的修改。系统实现并扩展了 PF_key 协议，使内核空间与用户空间的通信方便灵活。今后可适当运用 IP 压缩(IPComp)减少数据包的长度，将其应用到 IPSec VPN 中，以提高网络吞吐量，以及将 VPN 网关与包过滤防火墙相结合等，这些技术都有待于进一步研究。

参考文献

[1] The Linux Kernel Archives[Z]. (2007-03-15). http://www.kernel.org.

[2] Krnt S, Atkindon R. Security Architecture for the Internet Protocol[S]. RFC 2401, 1998.

[3] Mcdonald D, Metz C, Phan B. PF_KEY Key Management API, Version 2[S]. RFC 2367, 1998.

[4] 张全林, 李　勤. Linux 内核 2.6 版中 IPSec 实现的研究[J]. 信息工程大学学报, 2005, 6(3): 48-51.

[5] Bollapragada V, Khalid M. IPSec VPN 设计[M]. 袁国忠, 译. 北京: 人民邮电出版社, 2006: 14-16.

(上接第 108 页)

参考文献

[1] 王福豹, 史　龙, 任丰原. 无线传感器网络中的自身定位系统和算法[J]. 软件学报, 2005, 16(5): 857-868.

[2] Bergamo P, Mazzini G. Localization in Sensor Networks with Fading and Mobility[C]//Proceedings of the 13th IEEE Personal, Indoor and Mobile Radio Communications. [S. l.]: IEEE Press, 2002: 750-754.

[3] Hu Lingxuan, Evans D. Localization for Mobile Sensor Networks[C]// Proceedings of the 10th Annual International Conference on Mobile Computing and Networking. [S. l.]: ACM Press, 2004: 45-47.

[4] Baggio A, Langendoen K. Monte-Carlo Localization for Mobile Wireless Sensor Networks[C]//Proceedings of the 2nd International Conference on Mobile Ad-hoc and Sensor Networks. Hongkong, China: [s. n.], 2006: 317-328.

[5] Dil B, Dulman S, Havinga P J N. Range-based Localization in Mobile Sensor Networks[C]//Proceedings of the 3rd European Workshop on Wireless Sensor Networks. Zurich, Switzerland: [s. n.], 2006: 164-179.

[6] 王小平, 曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 1998.