文章编号：1005-8451 (2010) 11-0038-04

在Click平台上实现IPSec/ESP隧道通信

易 李，张宏科，周华春
（北京交通大学 电子信息工程学院，北京 100044）

摘 要：Click是一种模块化软件路由器的开发平台。为了增强Click路由器的安全性，本文在普通Click路由器的基础上，利用Click平台可灵活配置的特点，为普通Click路由器加载了IPSec/ESP模块，并进行了验证测试和性能分析。

关键词：Click；IPSec/ESP；隧道通信
中图分类号：TP393 文献标识码：A

Implementation of IPSec/ESP tunnel communication on Click platform

YI Li, ZHANG Hong-ke, ZHOU Hua-chun
(School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Click was a development platform for building modular router. In order to enhance the security of Click modular router, the purpose of this paper was to use its flexible configuration features, based on the general Click modular router, loading an IPSec/ESP module on it, and carried out its verification tests and performance analysis.

Key words: Click; IPSec/ESP; tunnel communication

Click是一种高效的开发模块化路由器的软件平台，以其灵活的配置和丰富的可扩展性受到研究人员的青睐。

随着互联网的不断发展，信息在交互中的安全性日益受到人们的重视。Click路由器在边界路由器、集群路由器等领域的实用价值使得增强Click

收稿日期：2010-04-02
基金项目：基金项目：国家高技术研究发展计划（863计划）；国家自然科学基金（60870015）
作者简介：易 李，在读硕士研究生；张宏科，教授。

4 技术方案

该网络办公系统采用B/S开发模式，仿Windows资源管理操作习惯。采用IIS6.0+.NET+SQL-Server2003架构，采用Web Services(DLL)、ActiveX技术，开发语言使用C#。ASP.NET是编译后再执行的，其运行速度更快。数据库访问技术在ASP.NET中是通过ADO.NET上的Managed-Provider所提供的应用程序编程接口，来实现数据源的数据访问，包括OLEDB和ODBC所支持的数据库。而ADO.NET的数据处理是采用3层以上结构，并且是面向无连接的模式。这样既保证了系统和数据库的安全、数据备份功能，又确保了数据安全。

ActiveX组件编译结束后生成OCX组件，将组件嵌入Web页面注册后即可浏览到程序。电子公章和电子签名有完善的加密措施，确保了文件的安全性与权威性。

基于安全和用户身份鉴定的考虑，用户名+密码+限定IP+身份锁的登录方式，确保了系统操作的安全。用户通过主页登录窗口输入用户名和密码进入网络工作平台，浏览主页中所有文件及相关内容，并进行日常办公。

5 结束语

随着Internet的迅速发展和普及，基于B/S体系结构开发应用程序成为主流方式。将.NET技术、ActiveX组件技术应用到B/S体系结构中，实现业务逻辑封装，提高软件的可重性和可维护性。本文介绍新乡桥工段铁路工务部门网络办公系统的设计及系统中涉及到的技术，供大家参考。

参考文献：

[1] 贾宗星. 基于工作流的协同办公系统的设计与实现[J]. 计算机时代，2009 (5).

路由器的安全性成为实际应用中的迫切需求[1]。本文在掌握Click软件体系结构，及IPSec相关知识的基础上，搭建具体的实验环境，实现了为普通Click路由器加载IPSec模块。

1 相关技术

1.1 Click路由器

在Click平台搭建网络应用模型的基本组件被称为element，每个基本组件只完成某一项简单的报文处理工作如计算校验和、封装IP头部等。组件之间以带方向的矢量来连接，矢量的方向指明了报文的传递方向。基本组件和矢量连接线共同构成了具有一定功能的网络应用模型。

组件间的连接关系有两种类型，第1种表示报文的传递由上行组件发起，此时下行组件为被动接受报文，第2种表示下行组件掌握报文传递的主动权，当下行组件需要处理报文时才会向上行组件请求报文，上行组件接到请求后，如果有待处理的报文就将其返回给下行组件。如图1，每个方框代表一个基本组件，FromDevice表示从网络端口读取报文，方框的矢量连接线代表了组件间的报文流向，矢量连接线的两个端点代表组件间的连接关系，实心表示上行组件发起报文传递，空心则相反。如FromDevice和Queue之间的报文传递由FromDevice发起。

图1 简单的Click应用模型

1.2 IPSec/ESP的相关知识

IPSec是由IETF定义的一个网络层安全标准，它是一套完整的多服务、多算法和多粒度的安全框架设计[2]。IPSec提供的最主要的服务是数据的保密性、完整性，以及针对重放攻击（即入侵者重放一次会话过程）的保护。IPSec为了支持以上特性，引入了两个新的协议头部，AH（Authentication Header，认证头）和ESP（Encapsulating Security Payload，封装安全净荷）以及两种使用模式，即传输模式（transport mode）和隧道模式（tunnel mode）。

由于AH并不支持数据加密功能，而且本文提出的方法是基于隧道模式的，这里只简单介绍ESP工作于隧道模式的情形。

ESP隧道封装后的IP数据包如图2，ESP提供了数据加密服务和数据的完整性检查。其中，加密的数据包括原始IP报头、TCP/UDP报头、应用层数据和ESP报尾。由于ESP报头是明文传输的，所以ESP头无需被加密。ESP认证报尾，包含了净荷数据的数字签名，需要被认证的数据包括了除新IP头部外的全部报文。

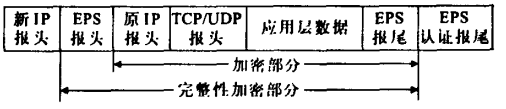

图2 ESP封装格式

2 IPSec/ESP在Click路由器上的实现

2.1 IPSec/ESP在Click平台上的设计方案

为了在Click路由器之间实现IPSec/ESP隧道通信，首先要搭建一个实验环境，拓扑如图3。路由器1和路由器2是两台加载了IPSec/ESP模块的Click路由器，用18.24.4.0网络来连接。主机1和主机2是两台普通工控机，这两台主机之间的通信会在路由器1和路由器2之间以ESP形式进行封装，本文主要完成了RFC2406中描述的封装安全载荷的通信过程。

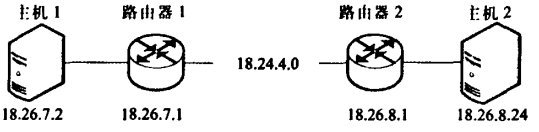

图3 实验环境

Click平台使用的版本号是1.7.0，与IPSec/ESP相关的组件由Dimitris Syrivelis在2006年开发完成。这些组件在最新版本的Click平台上运行时，会在路由表查询、加密密钥长度等方面出现错误。本文对这些错误进行了修改，并重新编译了Click。

2.2 在Click路由器上加载IPSec/ESP模块

根据IPSec/ESP的设计方案，在Click路由器上加载网络安全模块的具体方法如图4，这里只给出与IPSec/ESP相关的组件及它们的连接关系。

原理模型有几点需要说明：（1）图中每一个方框代表一个组件，组件名称后面是它的参数。（2）RadixIPsecLookup除了图中所示输入源，还有一个输入源用来接收经路由器处理过的外部网

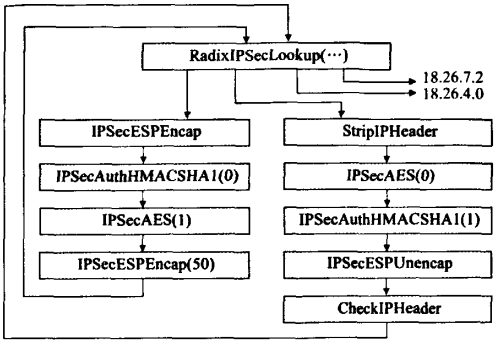

图4 IPSec/ESP在Click平台上的实现原理

络数据包。(3) 经RadixIPsecLookup的后两个输出端口发出的数据包，最终被分别转发至18.26.7.2主机和18.26.4.0网络。

以路由器1为例具体说明在ESP封装与解封装中各个组件的工作机制以及报文处理流程。当主机1发送一个目的地址为主机2的数据包时，路由器1首先接收到这个数据包，从RadixIPsecLookup开始数据包进入ESP模块进行处理。

2.2.1 ESP封装模块的处理流程

数据包从RadixIPsecLookup的第1个输出端口输出时，即进入了ESP封装流程。按照封装的顺序对每个组件的具体功能介绍如下：

RadixIPsecLookup：主要使用radix trie算法进行IP路由查询，并根据参数决定是否将报文送入ESP隧道，IPSec路由表在功能上类似一个特殊路由表，如果只对路由项设置目的IP地址，网关IP地址以及输出端口，这个组件可以完成普通IP路由查询功能，在ESP隧道通信中该组件则同时起着维护安全关联数据库的作用。当需要对某一条路由的报文进行ESP封装时，以下参数必须进行设置：SPI（Security Parameter Index，安全参数索引）值、128 bit的针对HMAC的认证密钥，128 bit的AES加密密钥，用于抗重放攻击的计数器和序列号，以及抗重放攻击的窗口大小。

IPsecESPEncap：为数据包封装ESP报头和ESP报尾，其中ESP报头包括32 bit的安全参数索引值，这个值在之前已经设置完毕，还有32 bit的序列号，这是一个单向递增的计数器，以及32 bit的初始化向量。ESP报尾包括填充项，填充项长度，下一个头部。由于加密采用的是AES方式，密钥长度为128 bit，所以ESP报尾要对数据部分进行填充，使加密部分扩展至128 bit的倍数。

IPsecAuthHMACSHA1：对数据报文的完整性hash验证并添加ESP认证报尾，此时参数为0。

IPsecAES：对数据包的原始报文和ESP报尾进行AES加密，此时参数为1。

IPsecEncap：为ESP封装后的报文再封装一个新的IP包头，这个新包头的目的IP地址应该是ESP隧道另一个端点的IP地址。这个组件的参数为50，表示报头协议域的值为50。

2.2.2 ESP解封装模块的处理流程

对于路由器2发来的数据包，路由器1首先对到达的数据包进行协议类型的检查，如果协议类型值为50，则认为该数据包为IPSec/ESP包，与ESP封装的流程类似，数据包从RadixIPsecLookup的第2个输出端口输出时，即进入了ESP解封装流程。这里还是按照解封装的顺序对每个组件的具体实现介绍如下：

RadixIPsecLookup：路由器1会从ESP头部中读取安全参数索引值和用于完整性检查的认证密钥，安全参数索引用于在哈希表中检索相应的安全关联（Security Association）属性，路由器1会根据这些属性对报文进行解密和完整性检查。

StripIPHeader：对ESP封装时的最后一个被添加的IP报头进行删除。

IPsecAES：对数据包的原始报文和ESP报尾进行AES解密，此时参数为0。

IPsecAuthHMACSHA1：对数据报文的完整性检查并删除ESP认证报尾，此时参数为1。

IPsecESPUnencap：删除数据包的ESP报头和报尾。

CheckIPHeader：检查IP包的版本号、头部长度、校验和等，如有错误则丢弃该数据包。

2.3 与IPSec/ESP相关的Click配置文件的编写

根据加载IPSec/ESP模块的方法，可以编写具体的Click配置文件，配置文件是用Click语言编写的，在Click平台上，这种语言是用来实现网络应用模型的高效的描述性语言。这里以路由器1为例给出与IPSec/ESP相关的配置文件，如下：

```
rt :: RadixIPsecLookup(18.26.4.24/32 - 1,
                       18.26.4.1/32 - 2,
                       18.26.7.0/24 - 3,
                       18.26.8.0/24 18.26.4.1 0 234
```

ABCDEFFF001DEFD2354550FE40CD708E 112233EE5566778888776655443322 11 300 64);

rt[0] ->IPsecESPEncap()
->IPsecAuthHMACSHA1(0)
->IPsecAES(1)
->IPsecEncap(50)-> [0]rt;

rt[1] -> StripIPHeader()
->IPsecAES(0)
->IPsecAuthHMACSHA1(1)
-> IPsecESPUnencap()
-> CheckIPHeader()-> [0]rt;

rt是RadixIPsecLookup的简称，rt的参数实际上是一个路由表，每个路由条目包括目的网络、网关和输出端口号，其中零号输出端口也就是第1个输出端口后边的参数，234代表安全参数索引值，ABCDEFFF001DEFD2354550FE40CD-708E代表了128 bit的HMAC认证密钥，112233EE-5566778888776655443322 11代表128 bit的AES加密密钥，300代表抗重放攻击计数器的初值，64代表抗重放攻击的窗口大小。以路由器1为例，RadixIPsecLookup的4个输出端口的数据包类型从左至右依次为：(1) 发送给18.26.8.0网络的数据包，需要被ESP封装。(2) 从路由器2发来的数据包，需要被解封装。(3) 发送给18.26.4.0网络的数据包，需要封装以太网头部，从相应的网络端口发出。(4) 发送给路由器2的数据包，同样需要封装以太网头部，经由网络端口发送至18.26.7.0网络。

rt[0]表示RadixIPsecLookup的第1个输出端口，它依次连接了IPsecESPEncap、IPsec-AuthHMACSHA1、IPsecAES以及IPsecEncap，最后数据包被传递至[0]rt，也就是RadixIPsec-Lookup的输入端口。

类似的，rt[1]表示RadixIPsecLookup的第2个输出端口，它依次连接的是StripIPHeader、IPsecAES、IPsecAuthHMACSHA1、IPsecESP-Unencap以及CheckIPHeader，数据包最终也被传递至[0]rt。

通过配置文件的编写，可以更进一步讨论数据包在IPSec/ESP模块内部的工作流程，以数据包在路由器1内被ESP封装的过程为例，数据包第1次进入RadixIPsecLookup时，将会被分配安全参数索引值，加密和认证密钥等ESP封装所必要的信息，之后数据包经由IPsecESPEncap，IPsec-AuthHMACSHA1等组件的处理，最后由IPsec-Encap为这个数据包封装新的IP头部，数据包会使用这个新的IP头部再次进入RadixIPsec-Lookup，这其实是一个二次查询路由表的过程的。ESP解封装的工作流程与封装相似，也需要两次查询路由表。

3 实验验证及性能分析

根据前文的叙述，在图3的基础上搭建了实验环境，并在18.26.4.0网络中使用抓包软件对ESP数据流进行捕获，截图如图5。在主机1上使用PING命令测试其与主机2的连通性和性能，如表1。

No	Time	Source	Destination	Protocol	Info
1	0.000000	18.26.4.24	18.26.4.1	ESP	ESP (SPI=0x000000ea)
2	0.000209	18.26.4.1	18.26.4.24	ESP	ESP (SPI=0x000000ea)
3	1.000848	18.26.4.24	18.26.4.1	ESP	ESP (SPI=0x000000ea)
4	1.001115	18.26.4.1	18.26.4.24	ESP	ESP (SPI=0x000000ea)
5	2.001675	18.26.4.24	18.26.4.1	ESP	ESP (SPI=0x000000ea)
6	2.001855	18.26.4.1	18.26.4.24	ESP	ESP (SPI=0x000000ea)
7	3.002538	18.26.4.24	18.26.4.1	ESP	ESP (SPI=0x000000ea)
8	3.002721	18.26.4.1	18.26.4.24	ESP	ESP (SPI=0x000000ea)

图5 ESP数据流

表1 性能测试表

	最短时间(ms)	平均时间(ms)	最长时间(ms)	方差(ms)
ESP	0.670	0.887	2.212	0.124
非ESP	0.596	0.664	0.810	0.045

表1对ESP封装和普通路由转发进行了性能比较，ESP封装的数据包的时延要更大一些，这是由ESP封装和加密算法共同造成。在平均时延上，两者相差并不大，可见在Click平台实现的ESP隧道通信的效率还很高。

4 结束语

本文在普通Click模块化路由器上，实现了IPSec/ESP模块的加载。在Click路由器间实现了ESP隧道通信，并对ESP隧道通信的连通性和性能进行了测试。IPSec极大地增强了普通Click路由器的安全性，这也使得Click路由器在边界路由器、集群路由器等领域的应用中更有实际意义。

参考文献：

[1] 傅　彬，王宝生，龚正虎. 软件集群路由器的研究与设计[C]. 2005年信息通信网络技术委员会年会征文，2005.

[2] Dimitris G. Syrivelis. IPsec support in Click Router[EB/OL]. http://www.read.cs.ucla. edu/click/docs/ipsec-doc. 2007/02/04.