Linux 下基于 DDNS 的动态 IPSec VPN 的研究①

Research on Dynamic IPSec VPN Based on DDNS under Linux

刘 权 陈蜀宇 (重庆大学 计算机学院 重庆 400030)

摘 要: IPSec VPN 是目前流行的基于 IP 的 VPN 部署技术。在实际应用中,如何实现动态 IP 环境下的 IPSec VPN 是一项关键技术。本文首先介绍 IPSec 的体系结构及 Linux 下 IPSec 的实现,紧接着针对动态 IP 环境,提出利用 DDNS 结合 Openswan 构建 Linux 下动态 IPSec VPN 的方案,最后给出该方案的具体实现。

关键词: IPSec VPN Linux DDNS Openswan 动态 IP

1 引言

随着企业的业务拓展和网络技术的迅速发展,越来越多的企业迫切需要将分支机构的局域网通过 Internet 连接起来,迅速安全地传递信息、共享数据。虚拟专用网(VPN)通过公共网络基础设施建立一个逻辑的、安全的专用连接隧道,提供专用网络服务。企业利用 VPN 组网技术,可以降低网络的运营费用,因此对 VPN 技术的需求日益强烈,VPN 技术在这一背景下迅速发展普及。

IPSec 是 IETF 制定的一套安全协议,采用 IPSec 构建 VPN 是一种主流的基于 IP 的 VPN 部署技术。IPSec VPN 通过建立安全隧道,对 IP 应用进行加密和认证。传统的 IPSec VPN 中,两端的网关均为固定 IP,即 IP 不发生变化。然而在实际应用中,由于 IP 地址并不充裕,拥有一个具有固定 IP 地址的线路,会耗费相当的费用。企业更愿意使用动态 IP 接入(如 ADSL、DHCP)。这样,当 IPSec VPN 网关的 IP 发生变化时,就会导致 VPN 失效。如何实现动态 IP 环境下的 IPSec VPN 是一项关键技术。本文在分析研究 IPSec 协议及 Linux 下 IPSec VPN 的实现基础之上,讨论动态 IP 环境下的 IPSec VPN 实现方案。

2 IPSec体系结构

IPSec 是由 IETF 开发的一套网络安全协议,可以“无缝”地为 IP (IPv4 和 IPv6)引入安全特性,提供可互操作的、高质量的、基于加密的安全服务。IPSec 旨在为 IP 分组提供安全服务,这些服务包括访问控制、数据完整性、身份验证、防止重放和数据机密性。在 RFC2401 中,对 IPSec 基本架构和基本部件做了如下定义:安全协议:验证报头(AH)和封装安全有效负载(ESP);密钥管理:ISAKMP、IKE、SKEME;算法:用于加密和身份验证[1]。

2.1 IPSec 安全协议

封装安全有效载荷(ESP)和验证报头(AH)是两种 IPSec 安全协议,用于为 IP 数据报提供这种安全。ESP 提供了机密性、数据完整性以及可选的数据来源验证和反重放服务。AH 提供无连接完整性、数据验证和可选的反重放服务,但不同于 ESP,它不提供机密性[2]。

2.2 IPSec 工作模式

IPSec 的这两种安全协议和 IPSec 的两种工作模式紧密相关,即传输模式和隧道模式。在传输模式下,在 IP 报头和高层协议报头之间插入一个 IPSec 报头(AH 或 ESP),这种模式只能用于 IP 端点和 IPSec 端点相同的情形。从 IPSec VPN 的角度看,即是保护两台主机之间的安全通信。在隧道模式下,原始 IP 分组被封装成一个新的 IP 数据报,并在内部报头和外部报头之间插入一个 IPSec 报头(AH 或 ESP)。由于这种封装包含一个外部 IP 报头,因此隧道模式可被用于在场点之间提供安全服务,场点代表了网关后面的 IP 节点。另外,这种模式也可用于远程终端主机连接到 IPSec 网关。

① 收稿时间:2008-09-16

2.3 IPSec 密钥管理

IPSec 使用 Internet 密钥交换协议(IKE)处理密钥管理问题。互联网安全关联密钥管理协议(ISAKMP)定义了协商、建立、修改和删除 SA 的过程和报文格式，它是一个与具体的密钥交换协议无关的框架协议。互联网密钥交换(IKE)协议是一个基于 ISAKMP 框架实现的协议，它为 IPSec 双方提供用于生成加密密钥和认证密钥的信息[3]。IKE 协议的一个主要功能就是 SA 的维护和管理。IKE 有两个不同的交换阶段，阶段 1 用于为 ISAKMP 通信双方建立一个 ISAKMP SA，然后用这个 ISAKMP SA 来为后续协商提供保护。阶段 2 用于为各种安全服务建立 IPSec SA。

2.4 安全关联和安全策略

安全关联(SA)是 IPSec 的一个基本部件，是通信对等方之间对安全参数的一种协定，例如建立安全连接所使用的加密算法。SA 是单向的，因此输入和输出的数据流需要独立的 SA。SA 是安全关联数据库(SADB)中的一个条目，SADB 包含双方协商的 IKE 或 IPSec 安全信息。SA 分为 IKE(ISAKMP) SA 和 IPSec SA。安全策略数据库(SPD)和安全联盟数据库(SADB)一样，都是构成 IPSec 的基础，分别用来管理安全策略信息和安全联盟数据。SPD 中的策略条目 SP 使用一个或多个选择符来指定其覆盖的 IP 数据流。每个条目 SP 都包含一个指示器，指出让与该策略匹配的数据流通过、丢弃还是进行 IPSec 处理[4]。

3 Linux下IPSec的实现

Linux 是著名的开放源代码的操作系统，同时也是优秀的网路操作系统，在 Linux 下实现 IPSec 是其主要实现方式[5]。Frees/WAN 在相当长的时间里是 Linux 下的主要的 IPSec 实现方式，现已停止开发。目前，Openswan 是 Linux 下对 IPSec 的最佳实现方式。

Openswan 是一个没有任何限制的完全开源的软件，它继承自 FreeS/WAN 项目，并增加了很多新的特点。在用户空间，Openswan 使用的工具是 Pluto，它是一个 IKE 守护进程。在内核空间，Openswan 使用的是继承自 FreeS/WAN 的 KLIPS 模块。自 Linux 2.5.47 开始的后续版本中，IPSec 已经成为内核的一部分，但 KLIPS 始终没有融入到 Linux 内核中。Linux 2.6 内核版本的 IPSec 实现基于 Kame 项目，Kame 项目是 Unix/BSD 家族的一部分。USAGI 项目使用 Kame 项目的 BSD 代码为基础，并集成 IPSec 到 Linux 内核中，形成了 NETKEY 模块，又称为“26sec”或“native”协议栈，用来替代 Openswan 的 KLIPS 模块。Openswan 非常灵活，可以用自身的 IPSec 内核堆栈 KLIPS，也可以用 Linux 下的 26sec。

IPSec VPN 最关键的问题之一是数据通过公共网络传输的安全性，保障数据安全的方法之一是对其进行加密。使用 IPSec 的安全特性，可对数据进行加密和验证，因此，构建 IPSec VPN 系统是 IPSec 的典型应用。Openswan 作为 Linux 下对 IPSec 的最佳实现方式，利用 Openswan 构建 Linux 下的 IPSec VPN 是一种快捷、安全的实现方案。

4 基于DDNS的动态IPSec VPN实现方案

4.1 静态 IPSec VPN 的问题

传统的 IPSec VPN 中，两端的网关均为固定 IP，即认为 IP 不发生变化，VPN 网关的配置不需要改变，构成静态的 IPSec VPN。然而由于 IP 地址并不充裕，对于一般的企业来说，拥有一个具有固定 IP 地址的线路，会耗费相当的费用。企业更愿意使用 ADSL 宽带线路，因此 IPSec VPN 常常面临动态 IP 的环境。当 VPN 网关为动态 IP 接入时，这个地址每次都是动态变化的，为保证 VPN 正常工作，就须解决 VPN 网关的 IP 寻址问题，也就是如何使动态的 IP 与固定的 VPN 网关相关联，“固化”它们的关系，这是实现动态 IP 环境下的 IPSec VPN 的关键技术。

4.2 动态 IPSec VPN 的方案

因此需要这样一种机制，使得 VPN 网关有一个唯一标识，并且由第三方管理这个标识。VPN 网关每次动态 IP 接入时，都会将自己新获得的 IP 地址通知第三方进行更新。对端 VPN 网关只需要每次访问这个第三方，查找这个唯一标识，获得对应的 IP 地址，然后就能与之进行通信。

Linux 下的动态域名服务 DDNS 正是能满足这种需求的机制，DDNS(Dynamic Domain Name Server)是将动态 IP 地址映射到一个固定的域名解析服务上，每次连接网络的时候，客户端通过信息传递把该主机的动态 IP 地址传送给服务器程序，实现动态域名解析。因此，让动态 IP 接入的 VPN 网关使用一个固定的域名，并上载所获得的 IP 到 DDNS 服务器后，VPN 隧道随即建立。在两端 IP 地址均不改变的情况下，

此 VPN 隧道一直提供服务，若有一方 IP 地址改变，隧道停止服务。IPSec VPN 系统在较短时间内作出响应，随即到 DDNS 服务器更新该域名对应的 IP 地址，对端 VPN 网关通过域名解析就可以知道更新后的 IP 地址，VPN 隧道再次建立。每个 DNS 记录都有一个 TTL(time-to-live)标识，DDNS 服务器根据这个标识决定每隔多长时间更新一次它的数据。那么，只需要把 TTL 值设置小一点，就可以保证域名的解析能及时生效。

Linux 下利用 BIND 软件可以方便地配置 DDNS 服务，Openswan 不仅支持用 IP 标识 VPN 网关，而且支持用域名标识 VPN 网关[6]。基于以上的研究和分析，本文针对动态 IP 环境，提出利用 DDNS 结合 Openswan 构建 Linux 下动态 IPSec VPN 的方案。

5 动态IPSec VPN的实现与测试

本文网络环境由 VMWare 虚拟机下的五台 Linux 虚拟主机组建，操作系统是 Fedora Core 7，内核版本：2.6.21。在两台虚拟主机上配置 Openswan 作为 IPSec VPN 网关，两台虚拟主机作为网关后的内网客户机，另外一台虚拟主机作为 DDNS Server。设置三个虚拟网段：10.10.0.0/24 模拟公网，192.168.1.0/24 是 Left 子网，192.168.2.0/24 是 Right 子网。五台 Linux 虚拟主机的子网掩码均为 255.255.255.0，网络参数如表 1 所示：

表 1　Linux 主机网络参数表

主机	eth0	eth1	默认网关
LeftGate	10.10.0.2	192.168.1.1	10.10.0.1
RightGate	动态 IP	192.168.2.1	10.10.0.1
LeftClient	10.10.0.20	192.168.1.2	192.168.1.1
RightClient	10.10.0.30	192.168.2.2	192.168.2.1
DDNS	10.10.0.3	-	10.10.0.1

为了简便，本文的 IPSec VPN 系统仅设置 RightGate 为动态 IP 接入，LeftGate 使用固定 IP。为此，设置的 10.10.0.0/24 虚拟网段提供 DHCP 服务，使 RightGate 的虚拟网卡 eth0 连接到网络时，获取到动态 IP。10.10.0.0/24 虚拟网段所属域为 vpnddns.net，RightGate 的主机名为 rightgate.vpnddns.net。网络环境如图 1 所示：

5.1 利用 Openswan 建立 VPN 网关间的 IPSec 隧道

目前 Openswan 的稳定版本为 2.x 系列，本文采用的版本是 Openswan-2.6.14。使用 26sec，不用给内核打 NAT-T 补丁就可以使用 NAT，为了方便，本文使用 Linux 自带的 26sec 协议栈。Openswan 有两种连接方式：Network-To-Network 方式和 Road Warrior 方式。本文采用 Network-To-Network 方式将两个网络连接成 VPN。Openswan 支持许多不同的认证方式，包括 RSA、PSK、XAUTH、X.509 证书方式。本文采用 RSA 数字签名(RSASIG)认证方式。

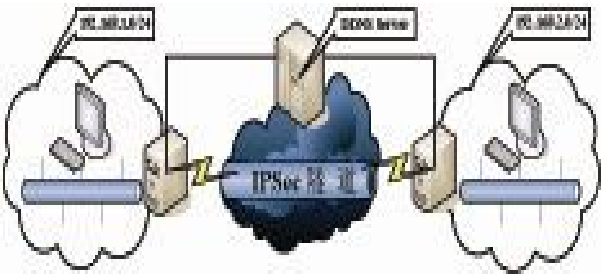

图 1　动态 IPSec VPN 网络环境示意图

5.1.1 打开路由转发

启动 Packet Forwarding 的功能，以允许数据包能从外部进入内部。编辑/etc/sysctl.conf，设置下面两项为：

```
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
```

执行 sysctl -p 使之生效。

5.1.2 Iptables 的 NAT 调整

不让 IP 伪装或 NAT 的数据包通过此 IPSec 隧道，在 LeftGate 和 RightGate 上分别执行以下命令：

```
root@leftgate ~]# iptables -t nat -A POSTROUTING -o eth0 -s 192.168.1.0/24 -d ! 192.168.2.0/24 -j MASQUERADE
[root@rightgate ~]# iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -d ! 192.168.1.0/24 -j MASQUERADE
```

5.1.3 安装 Openswan 源码包

```
[root@leftgate src]# tar zxvf openswan-2.6.14.tar.gz
[root@leftgate src]# cd openswan-2.6.14
[root@leftgate openswan-2.6.14]# make programs
[root@leftgate openswan-2.6.14]# make install
```

5.1.4 生成 RSA 密钥

生成 hostkey，写入 Openswan 配置文件 /etc/ipsec.secrets：

[root@leftgate ~]ipsec newhostkey --output /etc/ipsec.secrets

在 LeftGate 上获得公钥 leftrsasigkey：

[root@leftgate ~]ipsec showhostkey --left

rsakey AQO4hgB5K

leftrsasigkey=0sAQO4hgB5KK…

在 RightGate 上获得公钥 rightrsasigkey：

[root@rightgate ~]ipsec showhostkey --right

rsakey AQOcwXlJb

rightrsasigkey=0sAQOcwXlJb…

5.1.5 配置/etc/ipsec.conf

LeftGate 的配置如下：

conn subnet-to-subnet

left=10.10.0.2

leftsubnet=192.168.1.0/24

leftid=@Left

leftrsasigkey=0sAQO4hgB5KK…

leftnexthop=%defaultroute

right=rightgate.vpnddns.net

rightsubnet=192.168.2.0/24

rightid=@Right

rightrsasigkey=0sAQOcwXlJb…

rightnexthop=%defaultroute

auto=add

RightGate 的配置将”right=rightgate. Vpn-ddns.net”改为” right=%defaultroute”即可，其它配置均同 LeftGate。这是本文实现方案的关键，由于 RightGate 为动态 IP 接入，在 LeftGate 的配置文件中，只能用固定的域名标志 RightGate。而在 RightGate 的配置文件中，用默认路由地址标识自身。

5.1.6 启动 IPSec 隧道连接

在 LeftGate 或 RightGate 上执行下面的命令进行 Network-To-Network 连接，最后的输出行中出现”IPSec SA established”，说明 IPSec 隧道建立成功。

[root@leftgate ~]# ipsec auto --up subnet-to-subnet

5.2 DDNS 的实现

本文使用 BIND 搭建 DDNS 服务，版本是 BIND 9.4.0。BIND(伯克利互联网域名)是 DNS 协议的一种实现，在 Linux 下的主要配置文件是 /etc/named.conf。

5.2.1 在 DDNS Server 上生成密钥

执行 dnssec-keygen -a HMAC-MD5 -b 128 -n HOST rightgate，生成两个密钥文件 Krightgate.+157+28455.key 及 Krightgate.+157+28455.private：

[root@ddns ~]#cat Krightgate.+157+28455.key

rightgate. IN KEY 0 2 157 H6Qr0VCHXEV/E4Kdqx3EeQ==

5.2.2 配置/etc/named.conf

```
key "rightgate" {
    algorithm hmac-md5;
    secret " H6Qr0VCHXEV/E4Kdqx3EeQ==";
};
zone "vpnddns.net" {
    type master;
    file "named.vpnddns.net";
    update-policy {
    grant rightgate name rightgate. vpnddns.net. A;
    };
};
```

5.2.3 DDNS 客户端的更新

将 DDNS Server 端生成的 Krightgate.+157+28455.key 及 Krightgate.+157+28455.private 两个文件复制到 RightGate 的 /usr/local/ddns 目录，执行以下命令更新 IP：

[root@rightgate ddns]# nsupdate -k Krightgate.+157+28455.key

> server 10.10.0.2

> update delete rightgate.vpnddns.net

> update add rightgate.vpnddns.net 600 A

```
10.10.0.8
> send
```

在 DDNS Server 端的/var/named 目录产生一个文件 named.vpnddns.net.jnl，随着客户端的更新而发生变化。

5.2.4 DDNS 客户端的自动更新

当 VPN 网关重新连接网络时，IP 会发生变化，为次，需要让 RightGate 自动更新，实现动态的 DNS。编写以下脚本添加到 cron 让更新过程周期性地自动执行：

```
[root@ rightgate ~]# vi /usr/local/ddns/ddns_autoupdate.sh
#!/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH
basedir="/usr/local/ddns"
keyfile="$basedir"/"
Krightgate.+157+28455.key"
ttl=500
outif="ppp0"
hostname=" rightgate.vpnddns.net "
servername="10.10.0.3"
newip=`ifconfig "$outif" | grep 'inet addr' | awk '{print $2}' | sed -e "s/addr\://"`
tmpfile=$basedir/tmp.txt
cd $basedir
echo "server $servername" > $tmpfile
echo "update delete $hostname A " >> $tmpfile
echo "update add $hostname $ttl A $newip" >> $tmpfile
echo "send" >> $tmpfile
nsupdate -k $keyfile -v $tmpfile
```

5.3 动态 IPSec VPN 的测试

在 LeftClient 上执行 ping 192.168.2.2 或在 RightClient 上执行 ping 192.168.1.2，ping 命令执行期间，在 LeftGate 或 RightGate 上执行 tcpdump -i eth0。LeftClient 和 RightClient 可以互相 ping 通，且 tcpdump 得到下面的输出：

```
13:26:28.468728 IP 10.10.0.2 > rightgate.vpnddns.net: ESP (spi= 0xa7c7f 3dd, seq=0x74), length 132
13:26:28.469558 IP rightgate.vpnddns.net > 10.10.0.2: ESP (spi=0x5efe9669,seq=0x74), length 132
13:26:29.468579 IP 10.10.0.2 > rightgate.vpnddns.net: ESP (spi=0xa7c7f3dd,seq=0x75), length 132
13:26:29.469443 IP rightgate.vpnddns.net > 10.10.0.2: ESP (spi=0x5efe9669,seq=0x75), length 132
```

看到“ESP”标志，说明已经实现加密传输，两个子网中的客户机的通讯就像在一个局域网一样。重启 RightGate 网络服务，使其重新获取动态 IP，DDNS 自动更新 IP，IPSec VPN 重新建立连接。测试表明，本动态 IPSec VPN 实现方案是有效、可行的。

6 结束语

VPN 通过强密码认证或加密算法提供数据保护，满足企业组网的需求。IPSec VPN 作为流行的基于 IP 的 VPN 部署技术，常常面临动态 IP 的环境。本文利用 DDNS 结合 Openswan 实现 Linux 下的动态 IPSec VPN，动态 IP 接入的 VPN 网关监听 IP 变化，通过客户端脚本，DDNS 自动更新 IP，IPSec VPN 重新建立连接。本方案不仅有较小的系统响应时间，且能节约企业的费用，是一种有效、可行的动态 IPSec VPN 实现方案。

参考文献

1 Kent S, Atkinson R. Security Architecture for the Internet Protocol. RFC2401,1998:3－7.

2 Bollapragada V, Khalid M, Wainner S. IPSec VPN 设计. 北京:人民邮电出版社, 2006:21－28.

3 戴宗坤,唐三平.VPN 与网络安全.北京:电子工业出版社, 2002:191－200.

4 何宝宏.IP 虚拟专用网技术.北京:人民邮电出版社, 2002:113－116.

5 马士超,王贞松.IPSec 协议实现及其现状分析.计算机工程, 2006,32(22):107－110.

6 Wouters P, Bantoft K. Building and Integrating Virtual Private Networks with Openswan. Birmingham: Packt Publishing Ltd, 2006:90－91.