防火墙包转发和VPN 性能测试

Version 0.0.4 *

Meihui Fan
范 美 辉
<mhfan@hhcn.com>

Copyright © 2005 HHTech. Co., Ltd. †
All rights reserved.

华恒科技 版权所有

2005年9月1日

*revised version
†http://www.hhcn.com

目录

1 防火墙包转发性能测试

1.1 任务描述

搭建环境测试防火墙各个接口之间的双向小包转发性能。

1.2 调试记录

1. 现象描述 ：
   KUTE 的内核模块编译好之后不能正常加载：

   ```
   insmod: error inserting './kute_snd.ko': -1 Invalid module format
   ```

   原因分析 ：
   编译模块时所用的内核版本与正在运行的内核版本不一致；或者编译模块所用的编译器版本与编译正在运行的内核的编译器版本不一致。

   解决方法 ：
   重新用统一的编译器和内核代码编制内核和模块(KUTE)。

2. 现象描述 ：
   在网络连通的前提下，KUTE sender 发的包KUTE receiver 收不到，但tcpdump 可以捕捉到。

   原因分析 ：
   似乎发的包没有经过KUTE receiver 注册的协议层。

   解决方法 ：
   给内核打补丁，并在编译时定义宏`GRAB_EARLY`，让KUTE receiver （能够）更早地抢取UDP 包，而不需经过完整的协议栈。

3. 现象描述 ：
   当转发接口之一为PCI 口并且发包率为40 KPPS 以上时，FIREWALL 工作一段时间（几秒到几十妙）之后不再转发包，而且报错：

   ```
   eth0: Too much work at interrupt, status=0x4050
   ```

   原因分析 ：
   未明。

   解决方法 ：
   未解决。

4. 现象描述 ：
   在使用PCI 接口转发，在发包率过大时，有时还会报这种错误：
   ```
   IRQ LOCK: IRQ26 is locking the system, disabled.
   ```
   此时，PCI 的收包接口必须重启才能正常工作。

   原因分析 ：
   未明。

   解决方法 ：
   未解决。

5. 现象描述 ：
   可以PING 通，但KUTE sender 发的包，KUTE receiver 收不到。

   原因分析 ：
   FIREWALL 的两个转发端口其MAC 地址重复了。

   解决方法 ：
   修改转发接口的MAC 地址。

1.3 性能测试

1.3.1 测试环境

硬件

FIREWALL ：Intel Xscal IXP425, 64 MB SDRAM, NPE(2) & PCI i82559(2);

PC(2) ：Intel P4, 512 MB DDRAM, RTL8169;

SWITCH ：八口、千兆，双VLAN 模式；

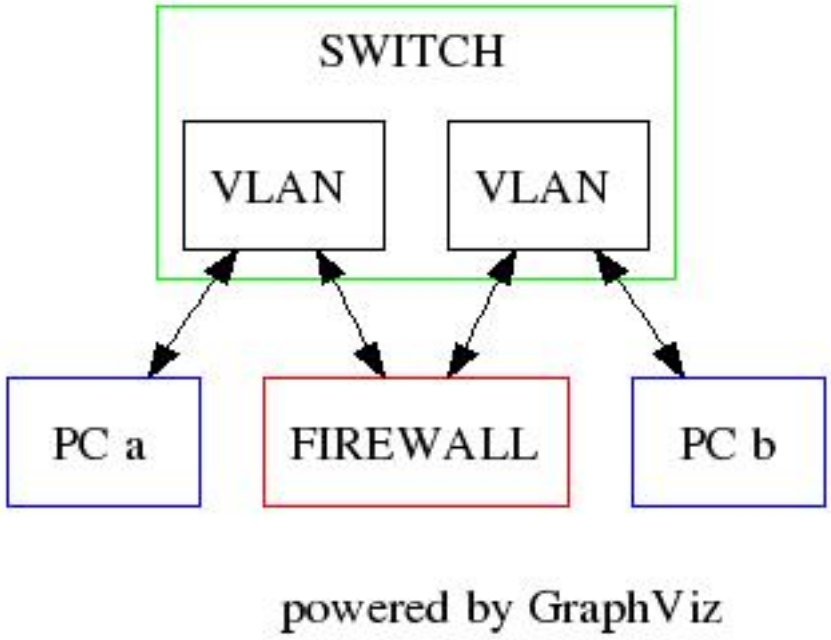
其中，两个VLAN 上分别有一个接口连到FIREWALL 的WAN 和LAN 接口上；而两台PC 机分别连到两个VLAN 上作为测试端，一台发包、另一台收包。如图：1.1。

软件

KERNEL(PC) ：v2.6.11.x

KUTE(PC) ：v1.2

KUTE 是一个基于2.6 系列内核的流量引擎，它以内核模块安装的方式来加载运行。



第 1.1 图: 防火墙性能测试环境

1.3.2 测试报告

测试命令

发送端 ：

```
PCa-$ ./kute_snd.sh -d 192.168.3.233 -l 64 -r 100000 -t 32
```

接收端 ：

```
PCb-$ ./kute_rcv.sh -w 100 -c 50 -t 30
```

测试结果

测试端口	(64B) PPS	(128B) PPS	(1480B) PPS
NPE -- NPE	54,012/54,236	56,286	8,241
PCI -- NPE	19,970/30,040	29,344	8,241
PCI -- PCI	21,773/19,587	20,231	8,240
NPE (IPT)	-/-	53,634	7,853
A--VPN--B	7,844/-	7,465	3,490

2 防火墙VPN 性能测试

2.1 任务描述

利用IPSEC 的FreeS/WAN 实现来构建两个防火墙之间net-to-net 的VPN 连接，并测试相应的包转发性能。

2.2 调试记录

1. 现象描述 ：
   修改FreeS/WAN 的配置文件使之支持net-to-net 的VPN 连接，重启IPSEC 模块之后，正确检测到ipsec0 接口，正确设置了路由表项，一切状态正常。但是，从Firewall A 发出的包始终不能到达Firewall B；反之亦然。不过，建立防火墙之间的host-to-host 连接之后，它们之间就能正常收发了。

   原因分析 ：
   IPSEC 方式的每一个VPN 连接是IP 层的处理方式，它只能处理所设置的源地址和目标地址，而不是针对接口的连接层/物理层等。所以，net-to-net 的连接只能处理net-to-net 的包，而其它的host-to-net/host-to-host 的包则被过滤。

   解决方法 ：
   因此，如果需要的话，应该为每一类的包建立相应的IPSEC/VPN 连接。

2.3 性能测试

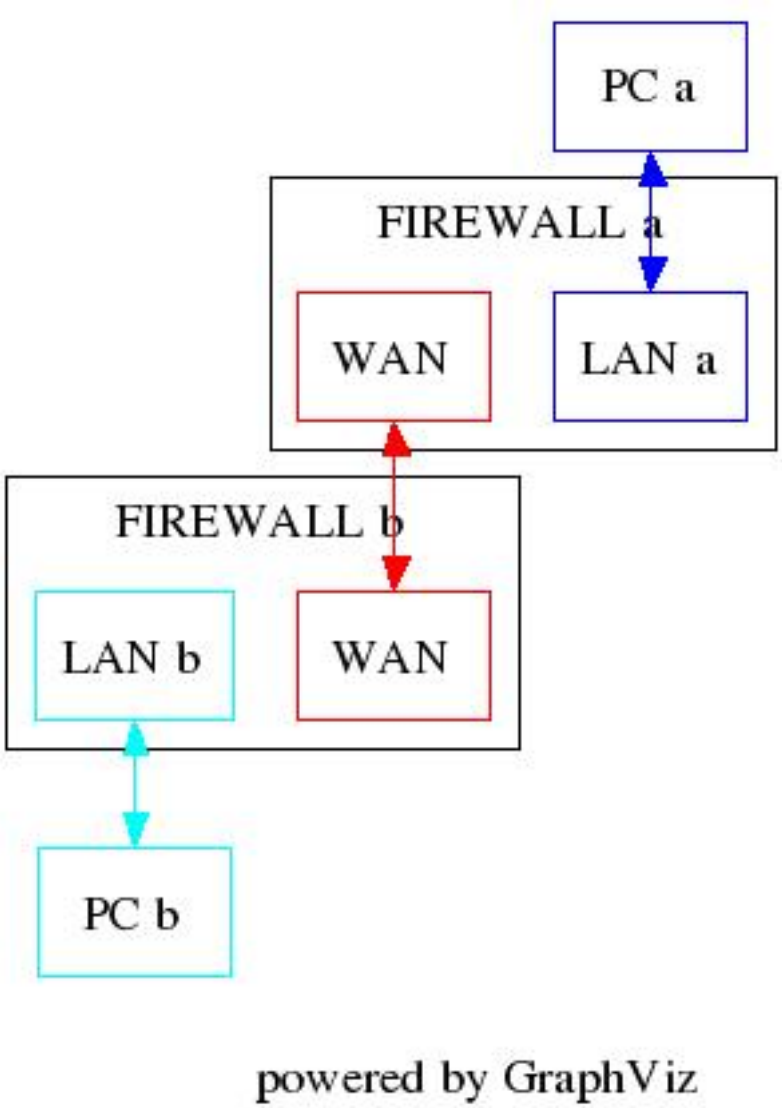
2.3.1 测试环境

硬件

FIREWALL(2) ：Intel Xscal IXP425, 64 MB SDRAM, NPE(2) & PCI i82559(2);

PC(2) ：Intel P4, 512 MB DDRAM, RTL8169;

这里，两个防火墙的外网接口(WAN)之间用双绞线直连；两台PC 分别接到两个防火墙的内部子网接口(LAN)。如图：1.1。



第 2.1 图: 防火墙IPSEC/VPN 测试环境

软件

FreeS/WAN(FW) ：v1.99

KERNEL(PC) ：v2.6.11.x

KUTE(PC) ：v1.2

2.3.2 测试报告

建立net-to-net 的VPN 连接

1. 设置防火墙之间的外网直连端口(WAN/ixp1)于同一网段，使之物理上能够直接连接；并且设置两个防火墙各自的内部子网端口(LAN/ixp0)分别于不同网段中：

```
FWa-$ ifconfig ixp0 192.168.1.1
FWa-$ ifconfig ixp1 192.168.3.1

FWb-$ ifconfig ixp0 192.168.2.1
FWb-$ ifconfig ixp1 192.168.3.2
```

   注意：要确保各个接口的以太网物理地址不同。

2. 编辑/etc/ipsec.conf 文件，使之包含如下内容：

```
config setup
interfaces="ipsec0=ixp1"
klipsdebug=none
plutodebug=none
plutoload=%search
plutostart=%search
uniqueids=yes

config %default
keyingtries=0
authby=rsasig
leftrsasigkey=0sAQPj......
rightrsasigkey=0sAQPj......

conn net-to-net
left=192.168.3.1
leftsubnet=192.168.2.0/24
leftnexthop=%direct
right=192.168.3.2
```

```
rightsubnet=192.168.1.0/24
rightnexthop=%direct
auto=start
```

注意：“auto”选项只须一端设置成“start”，另一端可以设置成“add”；其它的内容在两个防火墙中完全一样。

3. 启动FreeS/WAN IPSEC：

```
FWa-$ /path/to/ipsec setup --start
```

注意：如果配置文件中的“auto” 选项是设置为“add”的话，还需要加一步才会真正建立VPN 连接：

```
FWa-$ /path/to/ipsec auto --up net-to-net
```

4. 子网段的路由设置：

```
PCa-$ route add -net 192.168.3.0/24 dev eth0
PCa-$ route add -net 192.168.2.0/24 gw 192.168.3.1 dev eth0

PCb-$ route add -net 192.168.3.0/24 dev eth0
PCb-$ route add -net 192.168.1.0/24 gw 192.168.3.2 dev eth0
```

注意：这里假设子网中的各PC 机器与防火墙之间的连接接口是“eth0”。在实际的使用中一般就是直接设置默认网关：

```
PCa-$ route add default gw 192.168.3.1 dev eth0
PCb-$ route add default gw 192.168.3.2 dev eth0
```

5. 测试VPN 连接：

```
PCa-$ ping 192.168.2.54
FWa-$ /path/to/tcpdump -i ixp1
```

这里，可以从（物理的）网络接口上捕捉到IPSEC/ESP 协议的加密包传输：

```
11:16:32.046220 192.168.2.54 > 192.168.1.2:  ESP(spi=0x3be6c4dc,seq=0x3)
11:16:32.085630 192.168.1.2 > 192.168.2.54:  ESP(spi=0x5fdd1cf8,seq=0x6)
```

测试命令

发送端 ：

```
PCa-$ ./kute_snd.sh -d 192.168.2.54 -l 64 -r 40000 -t 40
```

接收端 ：

```
PCb-$ ./kute_rcv.sh -a 192.168.1.2 -w 100 -c 50 -t 30
```

测试结果

见第**??** 节，表1.3.2 最后一项。