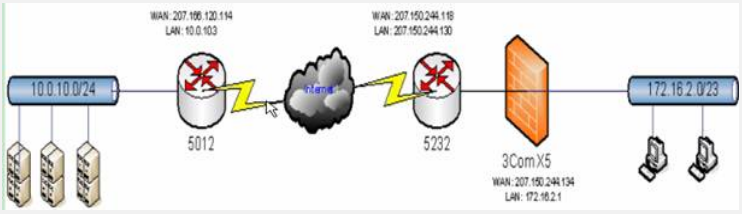
AR路由器与3Com X5 对接IPSec的经验案例

姚忆斌　2007-07-17 发表

AR路由器与3Com X5 对接IPSec的经验案例

一、组网：



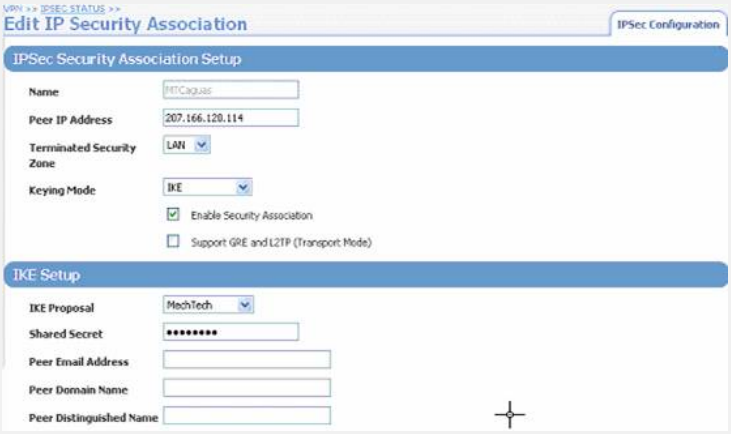
要求：客户要求图中R5012（AR28-10）与3Com X5设备之间建立IPSec隧道，以实现10.0.10.0/24网段与172.16.2.0/23网段之间通过公网进行安全通信。

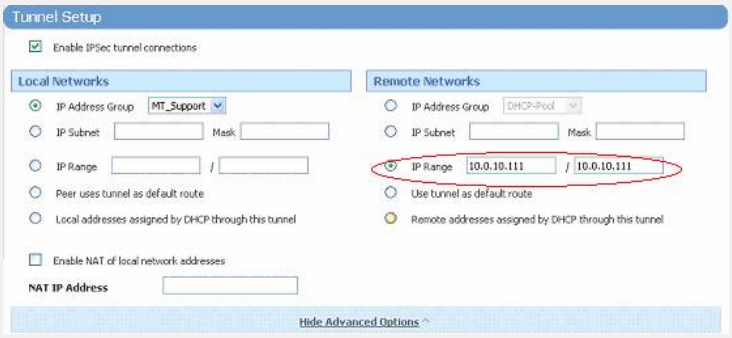
注：1、R5012和X5设备上使能NAT，R5232（AR28-31）并没有使能NAT，因此本组网中不存在NAT穿越的问题。

2、X5为3Com的IPS网关设备，可通过Web界面进行配置。

二、问题描述

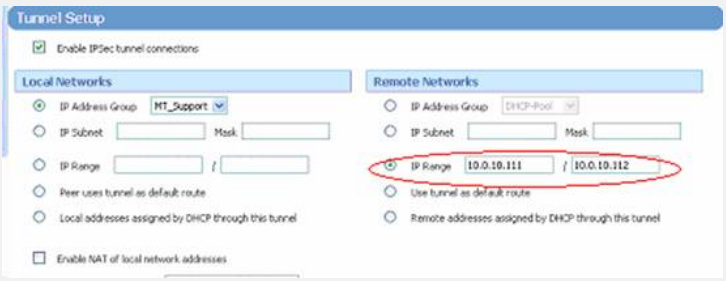
一线工程师现场调试时，发现如果X5设备后的私网只包含一台PC时，IPSec可以正常建立。X5上的主要配置如下：





图中红色标记处标识了在X5上定义了私网主机的地址，此时地址范围为10.0.10.111到10.0.10.111，即标识了一台主机。

由于客户的私网设备中不只包含了一台设备，于是将方案应用于网络中时对IP Range的设置进行了修改，修改后的配置如下：



此时地址范围为10.0.10.111到10.0.10.112，即标识了网络中有两台主机

相应的在R5012上的Security ACL配置也发生了变化：

更改前的配置:
acl number 3005
 rule 0 permit ip source 10.0.10.111 0 destination 192.168.1.150 0
更改后的配置:
acl number 3005
 rule 0 permit ip source 10.0.10.111 0 destination 192.168.1.150 0
 rule 1 permit ip source 10.0.10.112 0 destination 192.168.1.150 0
应用新的配置后发现IPSec隧道协商失败，具体现象是IKE SA协商成功，但是IPSec第二阶段协商失败，debug信息显示:
IKE packet dropped: (src addr: 207.166.120.114, dst addr: 207.150.244.134) with I_COOKIE d74a08192947d8cd and R_COOKIE 7
b7ab8dbc3a79202, because of ' **No IPSec policy found** ' from payload PROPOSAL.

三、过程分析
如上文所描述IPSec协商成功与IPSec协商失败之间唯一的配置差别就是ACL发生了变化。根据IPSec的协商原理，安全ACL正是IPSec第二阶段协商的重要内容，因此本问题可以基本断定与ACL的配置有关。
联想到在AR使用V3平台上安全策略视图下的ACL有三种配置方式（R0201版本后支持）:
 例: ipsec policy abc 10 isakmp
       security acl 3005
或security acl 3005 aggregation
或security acl 3005 per-session
       ike-peer 123
 proposal 456
即IPSec保护ACL下数据流的方式有如下三种:
standard方式：找到报文所匹配的ACL，为ACL下的每个规则建立一个隧道进行保护。
aggregation方式：找到报文所匹配的ACL，为ACL下的所有数据流建立一个隧道进行保护。
per-session方式：找到报文所匹配的ACL，建立一个隧道，该隧道仅保护与这个报文具有相同源和目的地址的报文。由于源和目的地址不同的报文很可能符合ACL下同一个规则，所以在per-session方式下，将会建立若干个隧道来保护ACL下的这个规则。
缺省情况下使用standard方式，据此我们怀疑本问题可能是X5设备的缺省方式与AR路由器不一致。一线工程是分别尝试使用aggregation方式和per-session方式后，发现故障现象依旧，于是否定了上述猜想。
只能反过头来分析两次debug信息，从中找线索。
在AR上的IPSec SA协商debug信息中发现了如下一段内容:
*0.440056809 MTCaguas IKE/8/DEBUG:message alloc: allocated 577c8e4
***0.440056909 MTCaguas IKE/8/DEBUG:message_recv: message 577c8e4** // **AR收到协商报文**
*0.440056958 MTCaguas IKE/8/DEBUG: ICOOKIE: 0xe6903637ca42d204
*0.440057057 MTCaguas IKE/8/DEBUG: RCOOKIE: 0x3b199ff67799eaea
*0.440057156 MTCaguas IKE/8/DEBUG: NEXT_PAYLOAD: HASH
*0.440057255 MTCaguas IKE/8/DEBUG: VERSION: 16
*0.440057354 MTCaguas IKE/8/DEBUG: EXCH_TYPE: QUICK_MODE
*0.440057453 MTCaguas IKE/8/DEBUG: FLAGS: [ENC]
*0.440057503 MTCaguas IKE/8/DEBUG: MESSAGE_ID: 0xda9109eb
*0.440057602 MTCaguas IKE/8/DEBUG: LENGTH: 292
*0.440057661 MTCaguas IKE/8/DEBUG:message dump: iovec 0:
 … …
*0.440060661 MTCaguas IKE/8/DEBUG:validate payload ID of message 577c8e4
*0.440060769 MTCaguas IKE/8/DEBUG: TYPE: 1
*0.440060770 MTCaguas IKE/8/DEBUG: DOI_DATA: 0x000000
*0.440060869 MTCaguas IKE/8/DEBUG: id information: type 1 proto 0 port 0
*0.440060918 MTCaguas IKE/8/DEBUG: id information: IPv4 address 192.168.1.150
*0.440061017 MTCaguas IKE/8/DEBUG:validate payload ID of message 577c8e4
***0.440061067 MTCaguas IKE/8/DEBUG: TYPE: 7**
*0.440061166 MTCaguas IKE/8/DEBUG: DOI_DATA: 0x000000
*0.440061265 MTCaguas IKE/8/DEBUG: id information: type 7 proto 0 port 0 // **ACL的type为 7**
*0.440061314 MTCaguas IKE/8/DEBUG:validate payload HASH of message 577c8e4

这时候对比调试成功时的信息：

*0.444168431 MTCaguas IKE/8/DEBUG:validate payload ID of message 5855844

*0.444168530 MTCaguas IKE/8/DEBUG: TYPE: 1

*0.444168530 MTCaguas IKE/8/DEBUG: DOI_DATA: 0x000000

*0.444168629 MTCaguas IKE/8/DEBUG: id information: type 1 proto 0 port 0

***0.444168678 MTCaguas IKE/8/DEBUG: id information: IPv4 address 192.168.1.150**

***0.444168777 MTCaguas IKE/8/DEBUG:validate payload ID of message 5855844**

***0.444168827 MTCaguas IKE/8/DEBUG: TYPE: 1 // ACL的type为1**

*0.444168926 MTCaguas IKE/8/DEBUG: DOI_DATA: 0x000000

*0.444169025 MTCaguas IKE/8/DEBUG: id information: type 1 proto 0 port 0

那么type为7的ACL是什么形式的呢？ 让我们看看其定义：

The **ID_IPV4_ADDR_RANGE** type specifies a range of IPv4 addresses,
represented by two four (4) octet values. The first value is the
beginning IPv4 address (inclusive) and the second value is the ending
IPv4 address (inclusive). All addresses falling between the two
specified addresses are considered to be within the list.

理解上述定义后我们不难发现type为7的ACL正是我们在X5上的配置方式，即使用一个起始地址和结束地址标识一些主机。

通过观察在X5设备上的LOG信息也证明了X5设备确实是发送了type为7的ACL：

2874 2007-06-19 14:03:50 INFO VPN 207.150.244.134:500

207.166.120.114:500 Initiator started IKE phase 2

2875 2007-06-19 14:03:50 INFO VPN 207.150.244.134:500

207.166.120.114:500 Peer phase 2 ID: **IPV4_ADDR_RANGE--10.0.10.111/10.0.10.112**

2876 2007-06-19 14:03:50 INFO VPN 207.150.244.134:500

207.166.120.114:500 Local phase 2 ID: IPV4_ADDR--192.168.1.150

四、结论：

由于AR路由器不能解析type 为7的ACL，导致IPSec协商失败，在V3平台上暂时不支持解析这种形式的ACL，若ACL type为1或4则解析正常。

下面介绍一下常见的三种ACL定义与类型对应关系：

1 IP subnet + 掩码的形式----------> ACL type 1

2 IP Group（地址池）---------->ACL type 4

3 IP地址范围（如1.1.1.1--1.1.10）------->ACL type 7

五、解决方法：

修改X5上ACL的配置方法，使其产生type为1或4的ACL，此时AR路由器能够正常解析并协商成功IPSec。配置方法如下：