小牛BLOG:www.ciscoedu.net

小牛QQ39358286

更多精彩尽在VIP-FTP

IPSEC VPN

IPSec

VPN---Virtual Private Network

虚拟专用网是一种业务,可以在共享的公共网络上提供安全可靠的连接通道.

按层次分的VPN:

二层VPN : ATM/Frame Relay/DDN/ISDN
三层VPN : IPSEC/GRE/L2TP
应用层VPN : Web VPN/SSL VPN

<安全的含义>

- 保证来源性 <===对源进行认证
- 保证完整性 <===在传输过程中,不允许对数据包进行修改
- 保证私密性 <===对数据包进行加密,又称为:数据的机密性.
- 不可否认性 <===不允许发送方抵赖,说自己没有传过

IPSEC VPN----IP security VPN:

	Site-to-site VPN	Remote access
VPN-enabled router	Primary role (full-fledged IOS)	Secondary ro
3000	Secondary role	Primary role (full-fledged ren access solutio
PIX Firewall	Security organization owns VPN solution	Enhance existing Firewall with the remote access so

1. Site-to-Site(又称为:Lan-to-Lan , 或者又称为:gateway-to-gateway)
   在公网上使用IPSEC VPN连接两个物理上不相连的局域网

2. Remote-access
   技术特点:
   适用于远程用户通过cisco VPN client 软件客户端拨号到中心站点的情况
   根据中心站点设备的不同分为Router remote VPN 和 PIX remote VPN
   关键技术点是1.5阶段的Xauth; VPN group; cisco VPN client

加密算法:
1.对称加密
   产生一条密钥(加密或解密都使用同一条密钥).
   加密速度快,加密后的文件紧凑(加密前后的文件大小差不多),适用于大量数据的加密.
   算法:DES/3DES/AES

2.非对称加密
   产生一对密钥(加密使用一条密钥(公钥),解密使用另一条密钥(私钥))
   加密速度慢,加密后的文件不紧凑(加密后的文件比加密前的文件大很多),适用于加密证书或KEY的管理.
   算法:RSA

DH Key Exchange ------- Diffie-Hellman 密钥交换(第一种公共密钥加密系统)

Data Integrity ------ 数据完整性
   HASH的算法: 1. MD5 128bit
   2. SHA-1 160bit

   HASH的特点: 1. 任意不同长度的输入,得到相同长度的输出.

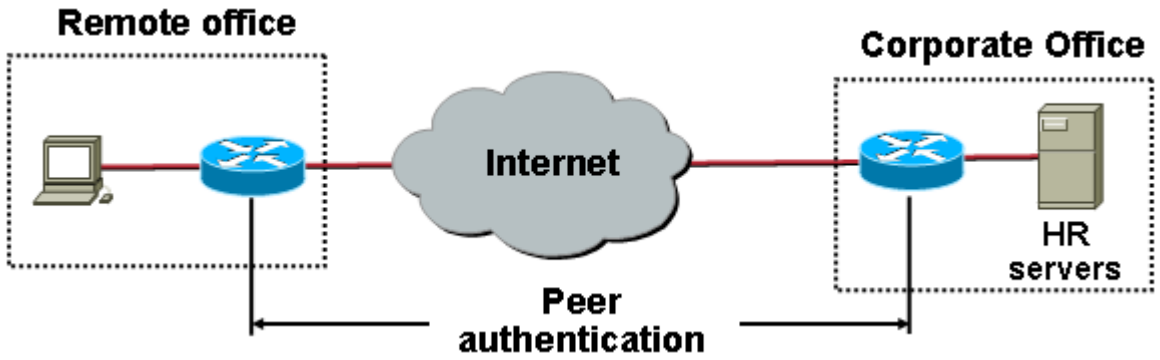
2. 只有完全相同的输入,才有完全相同的输出
3. 雪崩效应(只要有一点更改,就会有非常大的改变)

HMAC ----- HASH Message Authentication Code (利用HASH检验数据完整性)

"MAC" 是一个与Hash密切相关的名词，即信息鉴权码(Message Authority Code)。它是与密钥相关的Hash值，必须拥有该密钥才能检验该Hash值,只有密钥的拥有者可以计算出新的散列值。

Digital Signatures ----- 先用私钥加密,然后用公私解密.用于认证源.

Peer Authentication ----- 只要是验证,就要对比HASH



Peer authentication methods:

1. Pre-shared keys ------ 由双方预先设置好
2. RSA signatures
3. RSA encrypted nonces ----- 不需要CA

三种封装的协议:

1. AH-----Authentication Header(认证报头)

验证头部,一种安全协议,只是用来验证头部和防重发.不对实际用户数据部分加密.可配合ESP使用.AH使用IP协议51进行通信.AH是为IP数据报提供无连接的完整性和数据来源验证,并提供重放(replay)攻击保护.

使用AH时,永远不能穿越PAT设备(穿越PAT时,源IP地址会变)； 做HASH验证时:不计算TTL值；

Authentication Data(认证字段)包含在AH Header字段中.

8	16	32 bit
Next Header	Payload Length	Reserved
Security parameters index （SPI）		
Sequence Number Field		
Authentication Data （Variable）		

AH头部各字段含义如下：

• 下一头（8比特）：标识紧跟验证头的下一个头的类型。在传输模式下，将是处于保护的上层协议的值，如UDP或TCP的协议值。在通道模式下，将是值4，表示IP－in－IP(IPv4)封装或IPv6封装的41这个值。

• 载荷长度（8比特）：以32位字为单位的验证头的长度，再减去2。例如，缺省的验证数据字段的长度是96比特（3个32位字），加上3个字长的固定头，头部共6个字长，因此该字段的值为4。

• 保留（16比特）：为将来使用。

• 安全参数索引（32比特）：用于与外部IP头的目的地址一起标识一个安全关联。

• 序号（8比特）：单增的计数器值，用于提供抗重播功能。

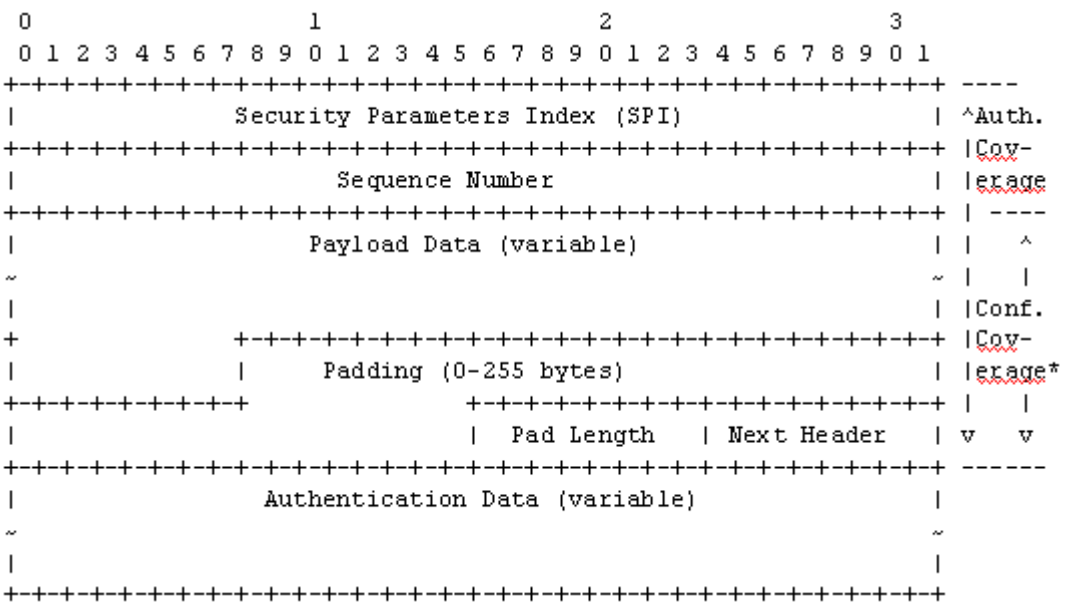
• 验证数据（可变）：该字段的长度可变（但应为32位字的整数倍），包含的数据有数据包的ICV（完整性校验值）或MAC。

RFC2402对AH头的格式、位置、验证的范围及进入和外出处理规则进行了描述。

2. ESP----Encapsulating Security Payload(封装安全有效载荷)

封装安全有效负载,一种提供数据加密的协议,同时支持验证和防重发功能,它完整封装用户数据,可独自使用或与AH配合使用.ESP使用IP协议50进行通信.

Authentication Data(认证字段)不包含在ESP Header字段中.



ESP头部各字段含义如下：

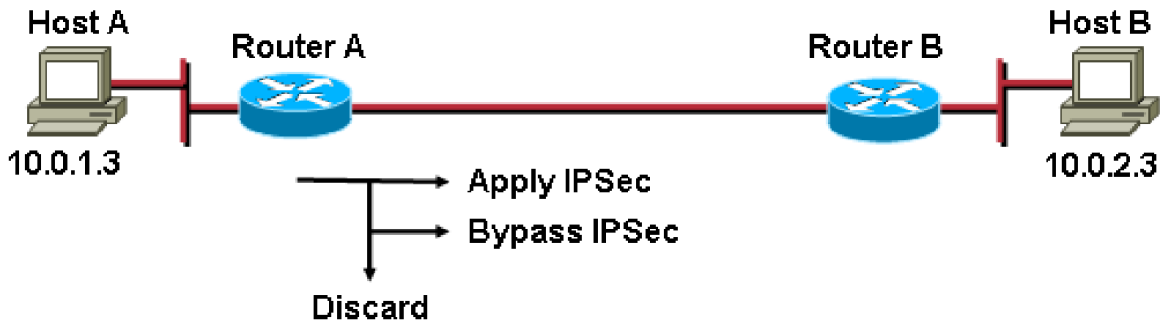
• SPI字段（Security Parameter Index(SPI)）：确定安全关联的安全参数索引,用于和IP头之前的目标地址以及协议一起标识一个安全关联。[32比特]
• 序列号字段（Sequence Number:）：用来提供反重放保护，跟验证报头中描述的一样。[32比特]
• 有效载荷数据（Payload Data）：传输层数据段（传输模式）或IP包（隧道模式），通过加密受到保护。也可在保护数据字段中包含一个加密算法可能需要用到的初始化向量（IV）。以强制实施的算法（DES-CBC）来说，IV是“受保护数据”字段中的第一个8位组。[可变]
• 填充字段(Padding: Extra bytes)：加密算法需要的任何填充字节。[0～9/10字节]
• 填充长度（Pad length）：包含填充长度字段的字节数[64 bit/块]
• 下一报头（Next Header）：通过标识载荷中的第一个头（如IPv6中的扩展头，或诸如TCP之类的上层协议头），决定载荷数据字段中数据的类型。
next header 取值: 1 for ICMP / 4 for IP-in-IP encapsulation / 6 for TCP / 17 for UDP
• 有效载荷数据（Authentication Data）：长度可变的字段（应为32位字的整数倍），用于填入ICV。ICV的计算范围为ESP包中除掉验证数据字段的部分。

ESP加密部分:Payload Data/Padding/Pad length/Next Header

RFC2406对ESP头的格式、位置、验证的范围及进入和外出处理规则进行了描述。

3. Encapsulating Security Payload + Authentication Header
包结构==> | Destination Source | AH HDR| ESP HDR| DATA|

Five steps of IPSec
1. Interesting Traffic --- The VPN devices recognize the traffic to protect
感兴趣流:用于VPN设备识别需要被保护的流量



如上图,IPSEC Tunnel 还没有建立起来:
(1) 当满足了感兴趣流定义的流量,那么将会丢弃.
(2) 没有满足感兴趣流定义的流量,对IPSEC Tunnel的建立与否不受到任何影响.

2. IKE Phase 1 --- The VPN devices negotiate an IKE security policy and

establish a secure channel

--- Authenticate the peers

--- Negotiate a bidirectional SA

--- Main mode or aggressive mode

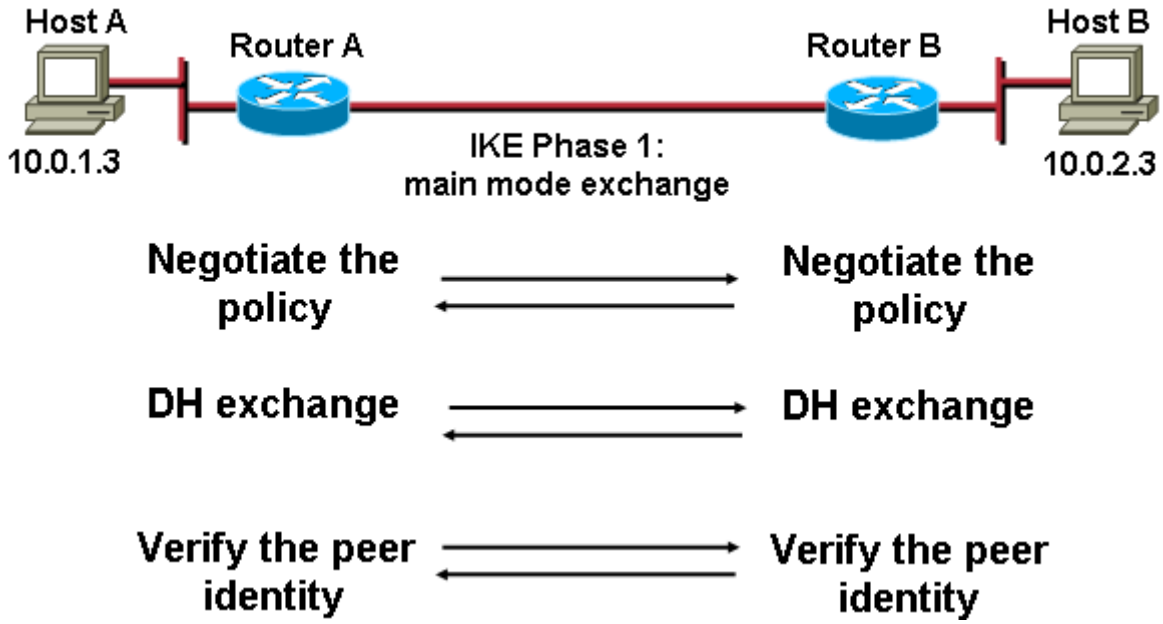
对认证双方PEER和为了加密一部分Main mode 和所有quick mode交换包,而协商一些参数.但是aggressive mode的包都没有加密.

1. 认证双方PEER
2. 产生KEY材料,这个用于产生为了实际加密数据,而用到的KEY
3. 在协商过程中: IKE使用用户数据报协议(UDP)端口500(通常用于源和目的双方)进行通信

所有的主模式和主动模式协商的信息都存在IKE或ISAKMP安全关联(SA)里.每两个PEER之间有一个安全关联.

Main mode (6 Messages) : Site-to-Site

aggressive mode (3 Messages) : Remote VPN 基于 Pre-share Key



上图中,只对第五步和第六步加密了(即Verify the peer identity),认证对方时使用"SKEYID_a".

3. IKE Phase 2 --- The VPN devices negotiate an IPSec security policy used to protect IPSec data

--- IPSec SAs/SPIs

--- Quick mode

在快速模式中,双方PEER协商IPSEC安全关联的属性值.用于加密(例如:ESP)两个主机之间的通信数据.如果启用PFS,将重新进行一次DH交换,在产生IPSEC数据加密KEY之前,交换新的KEY材料.

4. Data transfer --- The VPN devices apply security services to traffic

and then transmit the traffic

5. Tunnel terminated --- The tunnel is torn down

SA----security association(安全关联)

SA只是向它所承载的流量提供安全服务的一个连接.SA包含了有关要加密哪种数据流量、如何加密它、是否以及如何认证它、加密密钥及其多长时间刷新的信息.SA是单向的，故欲进行双向通信需建立两个SA（各为一个方向）.这些SA通过ISAKMP协商或可人工定义.

SA包括以下两个数据库:

1. Security Policy Database
   (1) Encryption Algorithm
   (2) Authentication Algorithm
   (3) Mode
   (4) Key lifetime

2. SA Database
   (1) Destination IP address <===用于新的IP包头和目的地址
   (2) SPI(安全参数索引) <===用于标识IPSEC通道
   (3) Protocol(ESP or AH)

IKE-----Internet Key Exchange(Internet密钥交换协议)

负责各种IPSEC选项的协商、认证通信的每一端(应用时包括公钥交换),以及管理IPSEC隧道的会话密钥.IKE使用用户数据报协议(UDP)端口500(通常用于源和目的双方)进行通信.

一种在Internet Security Association and Key Management Protocol (ISAKMP) 框架中使用Oakley和SKEME协议组的混合协议.IKE通常用来确定一个共享的安全策略和对需要KEY 的KEY服务的验证,在IPSEC流量能通过之前,先要对router/firewall/host 这些对等体进行身份验证.可以在双边手工输入预共享(pre-share)key或者通过CA获得KEY,通过双边协商双边获得统一IKE的SA,建立初步的安全通道,为接下来的IPSEC作准备.

IKE的功能:

1. Negotiating protocol parameters 协商协议参数
2. Exchanging public keys 交换公钥
3. Authenticating both sides 认证PEER
4. Managing keys after the exchange 管理交换完成的KEY

IKE是一个"元"(meta)协议: "ISAKMP" = "Oakley" + "SKEME"

1. ISAKMP----Internet Security Association and Key Management Potocol (Internet安全连接和密钥管理协议)

作用:定义了一个信息交换的体系架构,包括包的格式和分组在两个Peer之间的传送方式和状态.

2. Oakley ------ 一种KEY交换协议,它的一个基本机制就是Diffie-Hellman KEY交换算法

作用:提供了为在2个IPSec Peer 之间达成一种相同的加密密钥,而需要的一种基于模式的机制.用户从会话中得到加密密钥.

3. SKEME-----Security Key Exchange Mechanism(安全密钥交换机制)

作用:提供了以认证为目的而使用公钥加密的机制,用于认证 IKE SA 的两端.

IKE:Other Functions

DPD ------ Dead peer detection

用于探测对端的Peer是否还存在.

NAT traversal

Encapsulates IPSEC packet in UDP packet

因为IPSEC包为三层包是没有端口信息,所以要将IPSEC的包封装进UDP包的4500号端口里面

IPSEC的包分两种: AH (不支持 NAT Traversal)

ESP (支持 NAT Traversal)

所有主机的数据在没有穿越PAT设备时,源端口和目标端口都是4500号端口;当第一台主机数据穿越PAT设备时,源端口或目标端口号不发生改变,但当第二台主机数据穿越PAT设备时,源端口会随机更改.

SKEYID是从秘密材料中衍生出的字符串，只有某次交换中的活跃双方才知道。

SKEYID_a是ISAKMP SA用来验证消息所使用的密钥材料。 <====HASH时,使用

SKEYID_e是ISAKMP SA用来保护消息的机密性的密钥材料。 <====第一阶段时,使用

SKEYID_d是非ISAKMP安全联盟用来衍生出密钥所使用的密钥材料。 <===衍生出ESP的密钥材料{即:也会根据前面的"SKEYID_d"衍生出新的 "SKEYID_a"(已经没用了) / "SKEYID_e"(已经没用了) / "SKEYID_d"(衍生出ESP的密钥材料) }

Crypto Map

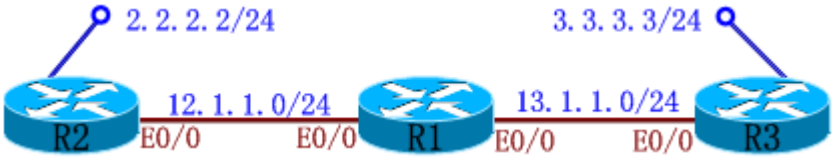
1. 静态的Crypto Map ----- 两端都使用静态的IP地址,例:Lan-to-Lan

2. 动态的Crypto Map ----- 其中一端使用动态的IP地址,例:Remote VPN

MAP的处理----对进来的数据包

是否感兴趣流	是否加密	有无MAP	action
N/A	是	有	解密
是	不	有	drop
是	不	没有	forward
N/A	是	没有	解密

实验1：Lan-to-Lan



IKE Phase I Policy:

协商策略(策略可以建立多个,发起方会将自己的所有策略发给对方,对方会按收到的序号最小的开始匹配；双方完全匹配的策略会由对方发回到发起方,表示确认,并使用此策略):

R2(config)#crypto isakmp policy 2 <===策略号只是本地有效(建议两端相同)

R2(config-isakmp)#authentication pre-share <====认证方法(Pre-share/rsa-encr/rsa-sig)

R2(config-isakmp)#hash md5 <====对协商包进行认证

R2(config-isakmp)#encryption 3des <===对协商的数据进行"3des"加密(两端一定要相同)

R2(config-isakmp)#group 2 <===两端组号一定要相同(不同的组:加密的位数也不相同)

R2(config-isakmp)#lifetime 60 <===多少秒后,重新协商(默认为:86400 second)

R2(config)#crypto isakmp key 0 wolf address 13.1.1.3 <===使用"Pre-share"认证方式,才要输入此命令

"0":密钥将以明文方式发向"13.1.1.3"

"wolf":密码(两端一定要一样)

R3(config)#crypto isakmp policy 3
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#hash md5
R3(config-isakmp)#encryption 3des
R3(config-isakmp)#group 2
R3(config)#crypto isakmp key 0 wolf address 12.1.1.2

IKE Phase II Policy:

定义转换集(本地所有的转换集要发往对方,让对方匹配自己的转换集,然后返回确认；可设置多个):

R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac

"cisco":transform-set的名字

"esp-des esp-sha-hmac":transform-set的加密和认证方式(选择"esp-null":不对数据加密)

R2(cfg-crypto-trans)#mode tunnel <===可选择 transport/tunnel(但Lan-to-Lan只能选择:tunnel模式)

定义感兴趣流:

R2(config)#access-list 101 permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
或者: R2(config)#ip access-list extended VPN
       R2(config-ext-nacl)#permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
定义MAP:
R2(config)#crypto map huawei 10 ipsec-isakmp
"huawei":定义MAP的名字
"10":序号
"ipsec-isakmp":可选择 ipsec-isakmp(自动)/ ipsec-manual(手动)
R2(config-crypto-map)#set peer 13.1.1.3   <===当满足了感兴趣流之后,和"13.1.1.3"建立peer
R2(config-crypto-map)#set transform-set cisco   <===调用名字为"cisco"的transform-set
或者: R2(config-crypto-map)#set transform-set cisco cisco2   <===后面可写多个"transform-set"
R2(config-crypto-map)#set pfs   <===可选择(group1/group2/group5),默认为"group1"
R2(config-crypto-map)#match address 101   <===匹配"101"(或者是"VPN")的感兴趣流

R3(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R3(cfg-crypto-trans)#mode tunnel
R3(config)#access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
R3(config)#crypto map huawei 10 ipsec-isakmp
R3(config-crypto-map)#set peer 12.1.1.2
R3(config-crypto-map)#set transform-set cisco
R3(config-crypto-map)#set pfs
R3(config-crypto-map)#match address 101

Apply VPN Configuration:
在接口上调用名字为"huawei"的MAP:
R2(config)#interface e0/0
R2(config-if)#crypto map huawei

R3(config)#interface e0/0
R3(config-if)#crypto map huawei

调试:
R2#debug crypto isakmp   <===查看第一阶段的IKE
R2#debug crypto ipsec   <===查看第二阶段的IKE

IPSEC中Lan-to-Lan的建立过程:
R2#ping 3.3.3.3 source 2.2.2.2
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
*Mar 1 00:55:40.767: IPSEC(sa_request): ,
(key eng. msg.) OUTBOUND local= 12.1.1.2, remote= 13.1.1.3, <==由于ping包的触发,开始建立IPSEC通道
local_proxy= 2.2.2.0/255.255.255.0/0/0 (type=4),
remote_proxy= 3.3.3.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= esp-des esp-sha-hmac (Tunnel),
lifedur= 3600s and 4608000kb,
spi= 0x5E39E570(1580852592), conn_id= 0, keysize= 0, flags= 0x400B

*Mar 1 00:55:40.767: ISAKMP: received ke message (1/1) <===IKE的第一阶段
*Mar 1 00:55:40.771: ISAKMP (0:0): SA request profile is (NULL)
*Mar 1 00:55:40.771: ISAKMP: local port 500, remote port 500
*Mar 1 00:55:40.771: ISAKMP: set new node 0 to QM_IDLE
*Mar 1 00:55:40.771: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa = 628E53B8
*Mar 1 00:55:40.771: ISAKMP (0:2): Can not start Aggressive mode, trying Main mode.
*Mar 1 00:55:40.775: ISAKMP: Looking for a matching key for 13.1.1.3 in default : success
*Mar 1 00:55:40.775: ISAKMP (0:2): found peer pre-shared key matching 13.1.1.3
*Mar 1 00:55:40.775: ISAKMP (0:2): constructed NAT-T vendor-07 ID
*Mar 1 00:55:40.775: ISAKMP (0:2): constructed NAT-T vendor-03 ID
*Mar 1 00:55:40.775: ISAKMP (0:2): constructed NAT-T vendor-02 ID
*Mar 1 00:55:40.775: ISAKMP (0:2): Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
*Mar 1 00:55:40.775: ISAKMP (0:2): Old State = IKE_READY New State = IKE_I_MM1

IKE第一阶段的第一个包的交换
*Mar 1 00:55:40.775: ISAKMP (0:2): beginning Main Mode exchange <==开始使用Main mode
*Mar 1 00:55:40.775: ISAKMP (0:2): sending packet to 13.1.1.3 my_port 500 peer_port 500 (I) MM_NO_STATE 发协包到对方PEER"13.1.1.3" 源端口:500 目标端口:500
*Mar 1 00:55:40.963: ISAKMP (0:2): received packet from 13.1.1.3 dport 500 sport 500 Global (I) MM_NO_STATE
*Mar 1 00:55:40.967: ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Mar 1 00:55:40.967: ISAKMP (0:2): Old State = IKE_I_MM1 New State = IKE_I_MM2

IKE第一阶段的第二个包的交换

```
*Mar  1 00:55:40.967: ISAKMP (0:2): processing SA payload. message ID = 0
*Mar  1 00:55:40.967: ISAKMP (0:2): processing vendor id payload
*M.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 12/12/12 ms
R2#ar  1 00:55:40.971: ISAKMP (0:2): vendor ID seems Unity/DPD but major
245 mismatch
*Mar  1 00:55:40.971: ISAKMP (0:2): vendor ID is NAT-T v7
*Mar  1 00:55:40.971: ISAKMP: Looking for a matching key for 13.1.1.3 in
default : success
*Mar  1 00:55:40.971: ISAKMP (0:2): found peer pre-shared key matching
13.1.1.3
*Mar  1 00:55:40.971: ISAKMP (0:2) local preshared key found
*Mar  1 00:55:40.971: ISAKMP : Scanning profiles for xauth ...
*Mar  1 00:55:40.971: ISAKMP (0:2): Checking ISAKMP transform 1 against
priority 2 policy
*Mar  1 00:55:40.971: ISAKMP:      encryption 3DES-CBC
*Mar  1 00:55:40.971: ISAKMP:      hash MD5
*Mar  1 00:55:40.971: ISAKMP:      default group 2
*Mar  1 00:55:40.971: ISAKMP:      auth pre-share
*Mar  1 00:55:40.971: ISAKMP:      life type in seconds
*Mar  1 00:55:40.975: ISAKMP:      life duration (basic) of 60
*Mar  1 00:55:40.975: ISAKMP (0:2): atts are acceptable. Next payload is
0 <===表示策略匹配协商完成
*Mar  1 00:55:41.143: ISAKMP (0:2): processing vendor id payload
*Mar  1 00:55:41.143: ISAKMP (0:2): vendor ID seems Unity/DPD but major
245 mismatch
*Mar  1 00:55:41.143: ISAKMP (0:2): vendor ID is NAT-T v7
*Mar  1 00:55:41.143: ISAKMP (0:2): Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Mar  1 00:55:41.143: ISAKMP (0:2): Old State = IKE_I_MM2  New State =
IKE_I_MM2

*Mar  1 00:55:41.147: ISAKMP (0:2): sending packet to 13.1.1.3 my_port
500 peer_port 500 (I) MM_SA_SETUP
*Mar  1 00:55:41.151: ISAKMP (0:2): Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Mar  1 00:55:41.151: ISAKMP (0:2): Old State = IKE_I_MM2 New State =
IKE_I_MM3
```

IKE第一阶段的第三个包的交换

```
*Mar  1 00:55:41.371: ISAKMP (0:1): purging node -1572961127
*Mar  1 00:55:41.383: ISAKMP (0:2): received packet from 13.1.1.3 dport
```

500 sport 500 Global (I) MM_SA_SETUP
*Mar 1 00:55:41.387: ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Mar 1 00:55:41.387: ISAKMP (0:2): Old State = IKE_I_MM3 New State = IKE_I_MM4

IKE第一阶段的第四个包的交换

*Mar 1 00:55:41.387: ISAKMP (0:2): processing KE payload. message ID = 0
*Mar 1 00:55:41.603: ISAKMP (0:2): processing NONCE payload. message ID = 0
*Mar 1 00:55:41.607: ISAKMP: Looking for a matching key for 13.1.1.3 in default : success
*Mar 1 00:55:41.607: ISAKMP (0:2): found peer pre-shared key matching 13.1.1.3
*Mar 1 00:55:41.607: ISAKMP (0:2): SKEYID state generated <===当第三、四个包完成后，就会产生SKEYID
*Mar 1 00:55:41.607: ISAKMP (0:2): processing vendor id payload
*Mar 1 00:55:41.607: ISAKMP (0:2): vendor ID is Unity
*Mar 1 00:55:41.611: ISAKMP (0:2): processing vendor id payload
*Mar 1 00:55:41.611: ISAKMP (0:2): vendor ID is DPD
*Mar 1 00:55:41.611: ISAKMP (0:2): processing vendor id payload
*Mar 1 00:55:41.611: ISAKMP (0:2): speaking to another IOS box!
*Mar 1 00:55:41.611: ISAKMP (0:2): Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Mar 1 00:55:41.611: ISAKMP (0:2): Old State = IKE_I_MM4 New State = IKE_I_MM4

*Mar 1 00:55:41.615: ISAKMP (0:2): Send initial contact
*Mar 1 00:55:41.615: ISAKMP (0:2): SA is doing pre-shared key authentication using id type ID_IPV4_ADDR
*Mar 1 00:55:41.615: ISAKMP (0:2): ID payload
        next-payload : 8
        type   : 1
        address   : 12.1.1.2
        protocol   : 17
        port   : 500
        length   : 12
*Mar 1 00:55:41.619: ISAKMP (2): Total payload length: 12
*Mar 1 00:55:41.619: ISAKMP (0:2): sending packet to 13.1.1.3 my_port 500 peer_port 500 (I) MM_KEY_EXCH
*Mar 1 00:55:41.623: ISAKMP (0:2): Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
*Mar 1 00:55:41.623: ISAKMP (0:2): Old State = IKE_I_MM4 New State = IKE_I_MM5

IKE第一阶段的第五个包的交换

```
*Mar  1 00:55:41.643: ISAKMP (0:2): received packet from 13.1.1.3 dport
500 sport 500 Global (I) MM_KEY_EXCH
*Mar  1 00:55:41.647: ISAKMP (0:2): processing ID payload. message ID = 0
*Mar  1 00:55:41.647: ISAKMP (0:2): ID payload
        next-payload : 8
        type         : 1
        address      : 13.1.1.3
        protocol     : 17
        port         : 500
        length       : 12
*Mar  1 00:55:41.647: ISAKMP (0:2): processing HASH payload. message ID =
0
*Mar  1 00:55:41.651: ISAKMP (0:2): SA authentication status:
        authenticated
*Mar  1 00:55:41.651: ISAKMP (0:2): SA has been authenticated with
13.1.1.3
*Mar  1 00:55:41.651: ISAKMP (0:2): peer matches *none* of the profiles
*Mar  1 00:55:41.651: ISAKMP (0:2): Input = IKE_MESG_FROM_PEER,
IKE_MM_EXCH
*Mar  1 00:55:41.651: ISAKMP (0:2): Old State = IKE_I_MM5  New State =
IKE_I_MM6
```

IKE第一阶段的第六个包的交换

```
*Mar  1 00:55:41.655: ISAKMP (0:2): Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Mar  1 00:55:41.655: ISAKMP (0:2): Old State = IKE_I_MM6  New State =
IKE_I_MM6

*Mar  1 00:55:41.659: ISAKMP (0:2): Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
*Mar  1 00:55:41.659: ISAKMP (0:2): Old State = IKE_I_MM6  New State =
IKE_P1_COMPLETE

*Mar  1 00:55:41.659: ISAKMP (0:2): received packet from 13.1.1.3 dport
500 sport 500 Global (I) MM_KEY_EXCH
*Mar  1 00:55:41.663: ISAKMP: set new node -1323939639 to QM_IDLE
*Mar  1 00:55:41.663: ISAKMP (0:2): processing HASH payload. message ID =
-1323939639
*Mar  1 00:55:41.667: ISAKMP (0:2): processing DELETE payload. message ID
= -1323939639
*Mar  1 00:55:41.667: ISAKMP (0:2): peer does not do paranoid keepalives.
```

```
*Mar  1 00:55:41.667: ISAKMP (0:2): deleting node -1323939639 error FALSE
reason "informational (in) state 1"
*Mar  1 00:55:41.667: ISAKMP (0:2): beginning Quick Mode exchange, M-ID
of -426260398
```

IKE的第二阶段的快速模式的开始

```
*Mar  1 00:55:41.775: IPSEC(key_engine): got a queue event...
*Mar  1 00:55:41.775: IPSEC(key_engine_delete_sas): rec'd delete notify
from ISAKMP
*Mar  1 00:55:41.779: ISAKMP (0:2): sending packet to 13.1.1.3 my_port
500 peer_port 500 (I) QM_IDLE
*Mar  1 00:55:41.779: ISAKMP (0:2): Node -426260398, Input =
IKE_MESG_INTERNAL, IKE_INIT_QM
*Mar  1 00:55:41.779: ISAKMP (0:2): Old State = IKE_QM_READY  New State =
IKE_QM_I_QM1
*Mar  1 00:55:41.779: ISAKMP (0:2): Input = IKE_MESG_INTERNAL, IKE_PHASE1
_COMPLETE
*Mar  1 00:55:41.783: ISAKMP (0:2): Old State = IKE_P1_COMPLETE  New
State = IKE_P1_COMPLETE

*Mar  1 00:55:42.287: ISAKMP (0:2): received packet from 13.1.1.3 dport
500 sport 500 Global (I) QM_IDLE
*Mar  1 00:55:42.295: ISAKMP (0:2): processing HASH payload. message ID =
-426260398
*Mar  1 00:55:42.295: ISAKMP (0:2): processing SA payload. message ID = -
426260398
*Mar  1 00:55:42.295: ISAKMP (0:2): Checking IPSec proposal 1
*Mar  1 00:55:42.295: ISAKMP: transform 1, ESP_DES
*Mar  1 00:55:42.295: ISAKMP:   attributes in transform:
*Mar  1 00:55:42.295: ISAKMP:      encaps is 1 (Tunnel)
*Mar  1 00:55:42.295: ISAKMP:      SA life type in seconds
*Mar  1 00:55:42.295: ISAKMP:      SA life duration (basic) of 3600
*Mar  1 00:55:42.295: ISAKMP:      SA life type in kilobytes
*Mar  1 00:55:42.295: ISAKMP:      SA life duration (VPI) of  0x0 0x46
0x50 0x0
*Mar  1 00:55:42.295: ISAKMP:      authenticator is HMAC-SHA
*Mar  1 00:55:42.295: ISAKMP:      group is 1
*Mar  1 00:55:42.299: ISAKMP (0:2): atts are acceptable.
*Mar  1 00:55:42.299: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 12.1.1.2, remote= 13.1.1.3,
    local_proxy= 2.2.2.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 3.3.3.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-sha-hmac  (Tunnel),
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x12
```

*Mar 1 00:55:42.299: IPSEC(kei_proxy): head = huawei, map->ivrf = , kei->ivrf =
*Mar 1 00:55:42.303: ISAKMP (0:2): processing NONCE payload. message ID = -426260398
*Mar 1 00:55:42.303: ISAKMP (0:2): processing KE payload. message ID = -426260398
*Mar 1 00:55:42.431: ISAKMP (0:2): processing ID payload. message ID = -426260398
*Mar 1 00:55:42.431: ISAKMP (0:2): processing ID payload. message ID = -426260398
*Mar 1 00:55:42.443: ISAKMP (0:2): Creating IPSec SAs
*Mar 1 00:55:42.443:   inbound SA from 13.1.1.3 to 12.1.1.2 (f/i) 0/ 0
        (proxy 3.3.3.0 to 2.2.2.0)
*Mar 1 00:55:42.443:   has spi 0x5E39E570 and conn_id 2000 and flags 13
*Mar 1 00:55:42.443:   lifetime of 3600 seconds
*Mar 1 00:55:42.443:   lifetime of 4608000 kilobytes
*Mar 1 00:55:42.443:   has client flags 0x0
*Mar 1 00:55:42.443:   outbound SA from 12.1.1.2   to 13.1.1.3   (f/i) 0/ 0 (proxy 2.2.2.0   to 3.3.3.0   )
*Mar 1 00:55:42.443:   has spi 235339588 and conn_id 2001 and flags 1B
*Mar 1 00:55:42.443:   lifetime of 3600 seconds
*Mar 1 00:55:42.443:   lifetime of 4608000 kilobytes
*Mar 1 00:55:42.443:   has client flags 0x0
*Mar 1 00:55:42.447: ISAKMP (0:2): sending packet to 13.1.1.3 my_port 500 peer_port 500 (I) QM_IDLE
*Mar 1 00:55:42.447: ISAKMP (0:2): deleting node -426260398 error FALSE reason ""
*Mar 1 00:55:42.447: ISAKMP (0:2): Node -426260398, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
*Mar 1 00:55:42.447: ISAKMP (0:2): Old State = IKE_QM_I_QM1 New State = IKE_QM_PHASE2_COMPLETE
*Mar 1 00:55:42.451: IPSEC(key_engine): got a queue event...   第二阶段完成后，就会建立SA
*Mar 1 00:55:42.451: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 12.1.1.2, remote= 13.1.1.3,
    local_proxy= 2.2.2.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 3.3.3.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-sha-hmac (Tunnel),
    lifedur= 3600s and 4608000kb,
    spi= 0x5E39E570(1580852592), conn_id= 2000, keysize= 0, flags= 0x13
<===证明SA已经建立完成

```
*Mar  1 00:55:42.451: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 12.1.1.2, remote= 13.1.1.3,
    local_proxy= 2.2.2.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 3.3.3.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-sha-hmac  (Tunnel),
    lifedur= 3600s and 4608000kb,
    spi= 0xE06FF44(235339588), conn_id= 2001, keysize= 0, flags= 0x1B
*Mar  1 00:55:42.455: IPSEC(kei_proxy): head = huawei, map->ivrf = ,
kei->ivrf =
*Mar  1 00:55:42.455: IPSEC(crypto_ipsec_sa_find_ident_head):
reconnecting with the same proxies and 13.1.1.3
*Mar  1 00:55:42.455: IPSEC(add mtree): src 2.2.2.0, dest 3.3.3.0,
dest_port 0

*Mar  1 00:55:42.455: IPSEC(create_sa): sa created,
  (sa) sa_dest= 12.1.1.2, sa_prot= 50,
    sa_spi= 0x5E39E570(1580852592),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 2000
*Mar  1 00:55:42.455: IPSEC(create_sa): sa created,
  (sa) sa_dest= 13.1.1.3, sa_prot= 50,
    sa_spi= 0xE06FF44(235339588),
    sa_trans= esp-des esp-sha-hmac , sa_conn_id= 2001
*Mar  1 00:55:51.371: ISAKMP (0:1): purging SA., sa=62E3C24C, delme=
62E3C24C
```

清除IPSEC中Lan-to-Lan的连接:

```
    R2#clear crypto isakmp    <===清除第一阶段的IKE
    R2#clear crypto sa

R2#show crypto isakmp sa

R2#show crypto ipsec sa
interface: Ethernet0/0
    Crypto map tag: huawei, local addr 12.1.1.2        <===本端的地址
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (2.2.2.0/255.255.255.0/0/0)
<====本端的感兴趣流
   remote ident (addr/mask/prot/port): (3.3.3.0/255.255.255.0/0/0)
<====远端的感兴趣流
    current_peer 13.1.1.3 port 500      <====远端的地址和端口号
      PERMIT, flags={origin_is_acl,}
     #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 3
```

```
      #pkts decaps: 3, #pkts decrypt: 3, #pkts verify: 3
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts compr. failed: 0
      #pkts not decompressed: 0, #pkts decompress failed: 0
      #send errors 0, #recv errors 0
       local crypto endpt.: 12.1.1.2, remote crypto endpt.: 13.1.1.3
       path mtu 1500, ip mtu 1500
       current outbound spi: 0x5C587C54(1549302868)
       inbound esp sas:
        spi: 0xECCDDAEC(3972913900)
          transform: esp-des esp-sha-hmac ,
          in use settings ={Tunnel, }
          conn id: 2001, flow_id: 1, crypto map: huawei
          sa timing: remaining key lifetime (k/sec): (4449567/3167)
          IV size: 8 bytes
          replay detection support: Y
          Status: ACTIVE
       inbound ah sas:
       inbound pcp sas:
       outbound esp sas:
        spi: 0x5C587C54(1549302868)      <===本地"outbound"的"spi" ,就是对方"inbound"的"spi"
          transform: esp-des esp-sha-hmac ,
          in use settings ={Tunnel, }
          conn id: 2002, flow_id: 2, crypto map: huawei
          sa timing: remaining key lifetime (k/sec): (4449567/3165)
                                        SA Lifetime: Data-based/Time-based
            假设两端时间不相同:由时间小的一端发起PING包是可通的,但由时间大的一端发起PING包是不通的.
          IV size: 8 bytes
          replay detection support: Y
          Status: ACTIVE
       outbound ah sas:
       outbound pcp sas:

R2#show crypto map
Crypto Map "huawei" 10 ipsec-isakmp
          Peer = 13.1.1.3
          Extended IP access list VPN
              access-list VPN permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
          Current peer: 13.1.1.3
          Security association lifetime: 4608000 kilobytes/3600 seconds
<===两端协商时间,使用小的时间("Security association lifetime"中的"seconds")
```

PFS (Y/N): Y　　<===重新做一次Diffie-Hellman交换

只有一端配置了PFS:

(1) 由配置了PFS的一端发起连接是可以成功的(但会显示"Attributes Not Supported",然后继续协商)

(2) 由没有配置了PFS的一端发起连接是不成功的

DH group: group1

Transform sets={

cisco,

}

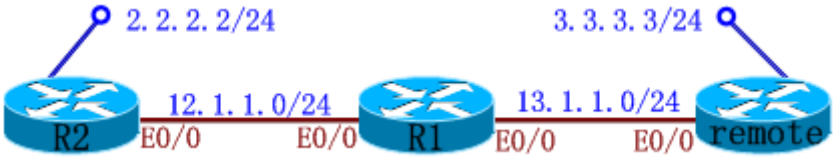
Interfaces using crypto map huawei:

Ethernet0/0

更改Security association lifetime:

```
R2(config)#crypto ipsec security-association lifetime seconds 600
R2(config)#crypto ipsec security-association lifetime kilobytes 10240

R2#show crypto engine connections active
  ID Interface            IP-Address      State  Algorithm
Encrypt  Decrypt
   1 Ethernet0/0          12.1.1.2        set    HMAC_MD5+3DES_56_C
0        0
2001 Ethernet0/0          12.1.1.2        set    DES+SHA
0       99
2002 Ethernet0/0          12.1.1.2        set    DES+SHA
99       0
```

实验2: Router Remote VPN ----- 只能是动态IP的一方发起连接(即,下例:"remote"方发起连接)



IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 2
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#group 2
R2(config)#crypto isakmp key 0 wolf address 0.0.0.0 0.0.0.0
```

<====由于是动态VPN,对端的地址不固定,所以写0.0.0.0

```
remote(config)#crypto isakmp policy 2
remote(config-isakmp)#authentication pre-share
remote(config-isakmp)#hash md5
```

```
remote(config-isakmp)#encryption 3des
remote(config-isakmp)#group 2
remote(config)#crypto isakmp key 0 wolf address 12.1.1.2
```

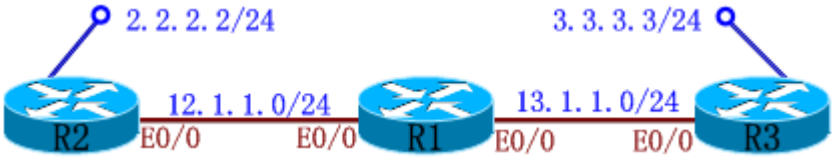
IKE Phase II Policy:

```
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(cfg-crypto-trans)#mode tunnel
R2(config)#access-list 101 permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
Static Crypto Map:
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 13.1.1.3
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address 101
--------------------------------------------------------------------------
remote(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
remote(cfg-crypto-trans)#mode tunnel
Dynamic Crypto Map:
remote(config)#crypto dynamic-map dynamap 10
remote(config-crypto-map)#set transform-set cisco
remote(config-crypto-map)#set pfs
remote(config)#crypto map QQ 10 ipsec-isakmp dynamic dynamap
```

Apply VPN Configuration:

```
R2(config)#interface e0/0
R2(config-if)#crypto map huawei
-----------------------------------
remote(config)#interface e0/0
remote(config-if)#crypto map QQ
```

实验3：Lan-to-Lan（新命令）



IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 2
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#group 2
R2(config)#crypto keyring L2LKEY
R2(conf-keyring)#pre-shared-key address 13.1.1.3 key wolf
R2(config)#crypto isakmp profile L2L
```

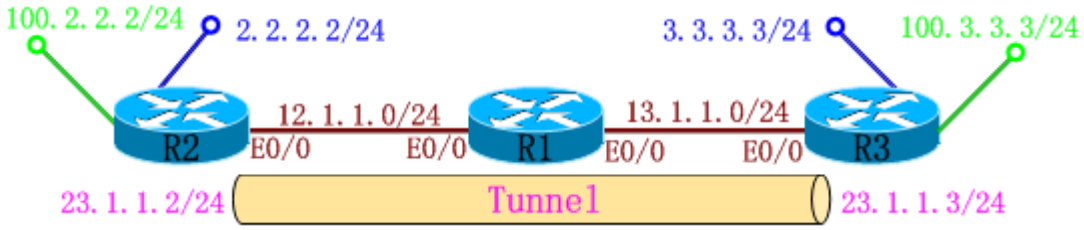
```
R2(conf-isa-prof)#match identity address 13.1.1.3
R2(conf-isa-prof)#keyring L2LKEY

IPSec Phase II Policy:
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(cfg-crypto-trans)#mode tunnel
R2(config)#access-list 101 permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 13.1.1.3
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address 101
R2(config-crypto-map)#set isakmp-profile L2L
R2(config-crypto-map)#reverse-route            <===选其一:反向路由注入/静态
(即:写了反向路由注入,可以不用写静态)

Apply VPN Configuration
R2(config)#interface ethernet 0/0
R2(config-if)#crypto map huawei
```

IPSec over GRE

IPSEC over GRE Configuration

技术特点:利用tunnel跑动态路由协议

实验 1 :




```
R2(config)#interface tunnel 23
R2(config-if)#ip address 23.1.1.2 255.255.255.0              <===起tunnel地址
R2(config-if)#tunnel source 12.1.1.2
R2(config-if)#tunnel destination 13.1.1.3
R2(config-if)#tunnel key 12345    <=== "tunnel key" 只是用于标识Tunnel,两端要对称,不是用于加密,在这可以不输入这条命令
R2(config-if)# router eigrp 90                    <===不用宣告连接Internet的接口
R2(config-router)#no auto-summary
R2(config-router)#network 2.2.2.0 0.0.0.255         <====宣告环回口网络
R2(config-router)#network 100.2.2.0 0.0.0.255          <====宣告内部网络
R2(config-router)#network 23.1.1.0 0.0.0.255                <====宣告tunnel地址
-------------------------------------------------
R3(config)#interface tunnel 23
R3(config-if)#ip address 23.1.1.3 255.255.255.0
R3(config-if)#tunnel source 13.1.1.3
R3(config-if)#tunnel destination 12.1.1.2
R3(config-if)#tunnel key 12345
R3(config-if)#router eigrp 90
R3(config-router)#no auto-summary
R3(config-router)#network 3.3.3.0 0.0.0.255
R3(config-router)#network 100.3.3.0 0.0.0.255
R3(config-router)#network 23.1.1.0 0.0.0.255
```

IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 1
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#group 2
R2(config)#crypto isakmp key 0 wolf address 3.3.3.3     <===一定要用环回口地址
```

```
R3(config)#crypto isakmp policy 1
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#hash md5
R3(config-isakmp)#encryption 3des
R3(config-isakmp)#group 2
R3(config)#crypto isakmp key 0 wolf address 2.2.2.2
```

IPSec Phase II Policy:

```
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(config)#access-list 101 permit ip 100.2.2.0 0.0.0.255 100.3.3.0 0.0.0.255
R2(config)#crypto map huawei local-address Loopback0        <===将"peer"改用"Loopback0"协商建立IPSEC通道(默认以物理口协商建立IPSEC通道)
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 3.3.3.3    <===可用公网接口地址,也可用环回口地址(与第一阶段设置无关)
```

以下四种情况每一次封装,先查路由表,再决定封装什么:

Peer设置为物理口,Map应用到公网接口时:

包结构: ...|tunnel source tunnel destination |GRE|source:100.2.2.2 destination:100.3.3.3|icmp...

由于应用到公网接口的Map,没有匹配到感兴趣流,所以没有加密直接发出.

Peer设置为物理口,Map应用到tunnel接口时:

包结构: ...|peer source peer destination |ESP|source:100.2.2.2 destination:100.3.3.3|icmp...

由于应用到tunnel接口的Map,匹配到感兴趣流,根据PEER的目标地址发出.(不经过tunnel,就出去了)

Peer设置为环回口,Map应用到公网接口时:

包结构: ...|tunnel source tunnel destination |GRE|source:100.2.2.2 destination:100.3.3.3|icmp...

由于应用到公网接口的Map,没有匹配到感兴趣流,所以没有加密直接发出.

Peer设置为环回口,Map应用到tunnel接口时:

包结构: ...|tunnel source tunnel destination |GRE|peer source peer destination |ESP|source:100.2.2.2 destination:100.3.3.3|icmp...

由于应用到tunnel接口的Map,匹配到感兴趣流,然后加密,根据PEER的目标地址,

继续查路由表,得出下一跳为Tunnel...(经过tunnel,从物理接口发出)

```
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address 101
--------------------------------------------------------------------------------
----
R3(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R3(config)#access-list 101 permit ip 100.3.3.0 0.0.0.255 100.2.2.0
0.0.0.255
R3(config)#crypto map huawei local-address Loopback0
R3(config)#crypto map huawei 10 ipsec-isakmp
R3(config-crypto-map)#set peer 2.2.2.2
R3(config-crypto-map)#set transform-set cisco
R3(config-crypto-map)#set pfs
R3(config-crypto-map)#match address 101
```

Apply VPN Configuration

```
R2(config)#interface ethernet 0/0
R2(config-if)#crypto map huawei
R2(config-if)#interface tunnel 23
R2(config-if)#crypto map huawei
------------------------------------------
R3(config)#interface tunnel 23
R3(config-if)#crypto map huawei
```

```
R3#show crypto engine connections active
  ID Interface            IP-Address      State  Algorithm
Encrypt  Decrypt
   1 Tunnel23             23.1.1.3        set    HMAC_MD5+3DES_56_C
0        0
2001 Tunnel23             3.3.3.3         set    DES+SHA
0        8
2002 Tunnel23             3.3.3.3         set    DES+SHA
8        0
------------------------------------------------------------
R2#show crypto isakmp sa
dst             src             state          conn-id slot status
1.1.1.1         2.2.2.2         QM_IDLE              1    0 ACTIVE
----------------------------------
R2#show crypto isakmp peers
Peer: 1.1.1.1 Port: 500 Local: 2.2.2.2
```

```
 Phase1 id: 1.1.1.1
--------------------------------
R2#show crypto ipsec sa
interface: Ethernet0/0
    Crypto map tag: cisco, local addr 2.2.2.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.1.2.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.1.1.0/255.255.255.0/0/0)
   current_peer 1.1.1.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 2.2.2.2, remote crypto endpt.: 1.1.1.1
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x12D1DDFE(315743742)

     inbound esp sas:
      spi: 0xC2686DB7(3261623735)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: 1, crypto map: cisco
        sa timing: remaining key lifetime (k/sec): (4386784/3492)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x12D1DDFE(315743742)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: 2, crypto map: cisco
        sa timing: remaining key lifetime (k/sec): (4386784/3490)
        IV size: 8 bytes
        replay detection support: Y
```

```
        Status: ACTIVE

      outbound ah sas:

      outbound pcp sas:

interface: Tunnel21
    Crypto map tag: cisco, local addr 2.2.2.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.1.2.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.1.1.0/255.255.255.0/0/0)
   current_peer 1.1.1.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 2.2.2.2, remote crypto endpt.: 1.1.1.1
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x12D1DDFE(315743742)

     inbound esp sas:
      spi: 0xC2686DB7(3261623735)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2001, flow_id: 1, crypto map: cisco
        sa timing: remaining key lifetime (k/sec): (4386784/3488)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x12D1DDFE(315743742)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: 2, crypto map: cisco
```

```
    sa timing: remaining key lifetime (k/sec): (4386784/3487)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

outbound ah sas:

outbound pcp sas:
```

GRE over IPSec

GRE over IPSEC Configuration

IPSec-Over-GRE和GRE-Over-IPSec方式配置上的区别为：

	GRE-Over-IPSec	IPSec-Over-GRE
ACL定义	GRE数据流	内网数据流
IKE peer中指定的remote-address	对方公网地址	对方GRE tunnel地址
应用端口	公网出口	GRE tunnel上

技术特点:

IPSec (ESP) tunnel only IP unicast traffic

GRE encryption non-ip and ip multicast or broadcast packets into ip unicast packets

Using a GRE tunnel inside an ipsec tunnel uses only three SA (at maximum)

GRE---Generic Routing Encapsulation

GRE是一个三层协议,无连接,没有安全性,支持的协议:IP / IPX / Apple Talk

Tunnel Mode 包结构:

| IP | ESP | IP | GRE | IP | TCP | Data | ESP |
|<=== Encrypted Payload ===>|

Transport Mode 包结构:

| IP | ESP | GRE | IP | TCP | Data | ESP |
|<=== Encrypted Payload ===>|

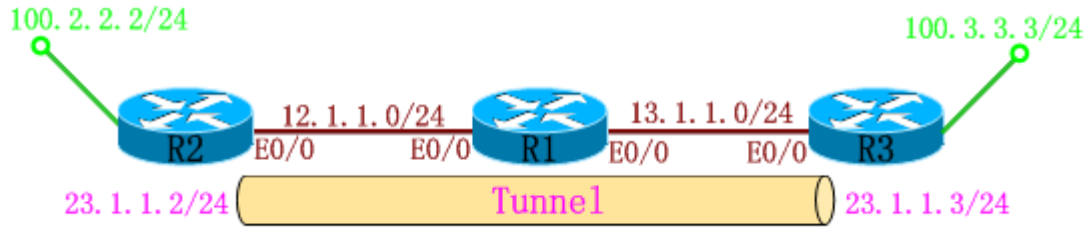
实验 1 :

本实验Tunnel Mode的包结构:

..| Peer source Peer destination| ESP| GRE的源地址 GRE的目标地址| GRE | 源IP 目标IP | data | ESP |..

本实验Transport Mode 包结构:

由于"Peer source Peer destination"(加密点)等于"GRE的源地址 GRE的目标地址"(通信点),所以包结构更改为: ...| GRE的源地址 GRE的目标地址| ESP| GRE | 源IP 目标IP | data | ESP |...



老命令:

起Tunnel:

```
R2(config)#interface tunnel 23
R2(config-if)#ip address 23.1.1.2 255.255.255.0          <===起tunnel地址
R2(config-if)#tunnel source 12.1.1.2
R2(config-if)#tunnel destination 13.1.1.3
-----------------------------------------
R3(config)#interface tunnel 23
R3(config-if)#ip address 23.1.1.3 255.255.255.0
R3(config-if)#tunnel source 13.1.1.3
R3(config-if)#tunnel destination 12.1.1.2
```

宣告:

```
R2(config-if)# router eigrp 90                    <===不用宣告连接Internet的接口
R2(config-router)#no auto-summary
R2(config-router)#network 100.2.2.0 0.0.0.255          <====宣告内部网络
R2(config-router)#network 23.1.1.0 0.0.0.255              <====宣告tunnel地址
-----------------------------------------------
R3(config-if)#router eigrp 90
R3(config-router)#no auto-summary
R3(config-router)#network 100.3.3.0 0.0.0.255
R3(config-router)#network 23.1.1.0 0.0.0.255
```

IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 1
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
```

```
R2(config-isakmp)#group 2
R2(config)#crypto isakmp key 0 wolf address 13.1.1.3          <===使用物理口地址
-----------------------------------------------------
R3(config)#crypto isakmp policy 1
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#hash md5
R3(config-isakmp)#encryption 3des
R3(config-isakmp)#group 2
R3(config)#crypto isakmp key 0 wolf address 12.1.1.2

IPSec Phase II Policy:
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
模式:
R2(cfg-crypto-trans)#mode tunnel              <===使用"tunnel"模式
或者:    R2(cfg-crypto-trans)#mode transport     <=== 使用"transport"模式.(只有"Peer source Peer destination"等于"GRE的源地址 GRE的目标地址"的特例中,才能使用,且只能在25系列路由器上做)
R2(config)#ip access-list extended gre
R2(config-ext-nacl)#permit gre any any   <===对条件可以抓的更细(any:可换成GRE的SOURCE/DESTINATION)
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 13.1.1.3          <===使用物理口地址
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address gre
-----------------------------------------------
R3(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R3(cfg-crypto-trans)#mode tunnel
R3(config)#ip access-list extended gre
R3(config-ext-nacl)#permit gre any any          <===对条件可以抓的更细
R3(config)#crypto map huawei 10 ipsec-isakmp
R3(config-crypto-map)#set peer 12.1.1.2
R3(config-crypto-map)#set transform-set cisco
R3(config-crypto-map)#set pfs
R3(config-crypto-map)#match address gre

Apply VPN Configuration
R2(config)#interface ethernet 0/0
R2(config-if)#crypto map huawei
-----------------------------------
R3(config)#interface ethernet 0/0
R3(config-if)#crypto map huawei
```

新命令：不需要感兴趣流,不需要MAP,不需要set peer

...　　<===之前的都一样

IPSec Phase II Policy:

R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac

R2(config)#crypto ipsec profile GREPRO　　<===只有26系列以上路由器才支持

R2(ipsec-profile)#set transform-set cisco

Apply VPN Configuration

R2(config)#interface tunnel 23

R2(config-if)#tunnel protection ipsec profile GREPRO

R2#show crypto ipsec sa　　<===可以查看协商成"transport"模式

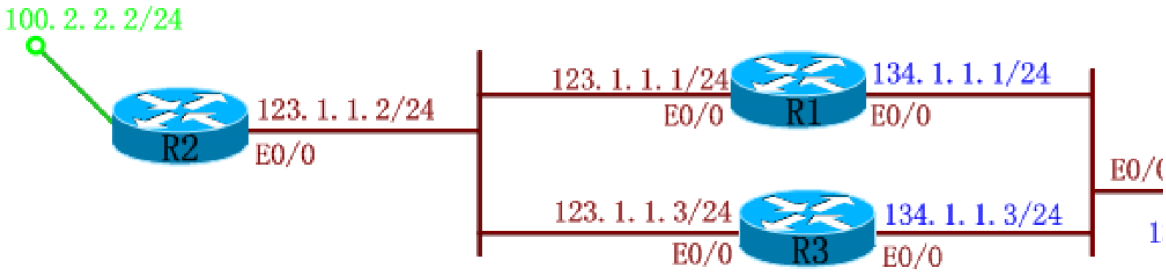
Backup Peer

Geographic HA using ipsec Backup peers

技术特点:

通过设置多个Peer的方式,Client能够自动切换Server的地址.
Peer是一个一个尝试,从上往下的顺序进行.

实验1:



IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 1
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#group 2
R2(config)#crypto isakmp keepalive 20  3     <===当第一个PEER断掉后,DPD开始侦测{过了60(20*3)秒后,开始切换}
R2(config)#crypto isakmp key 0 wolf address 123.1.1.1
R2(config)#crypto isakmp key 0 wolf address 123.1.1.3
....
```

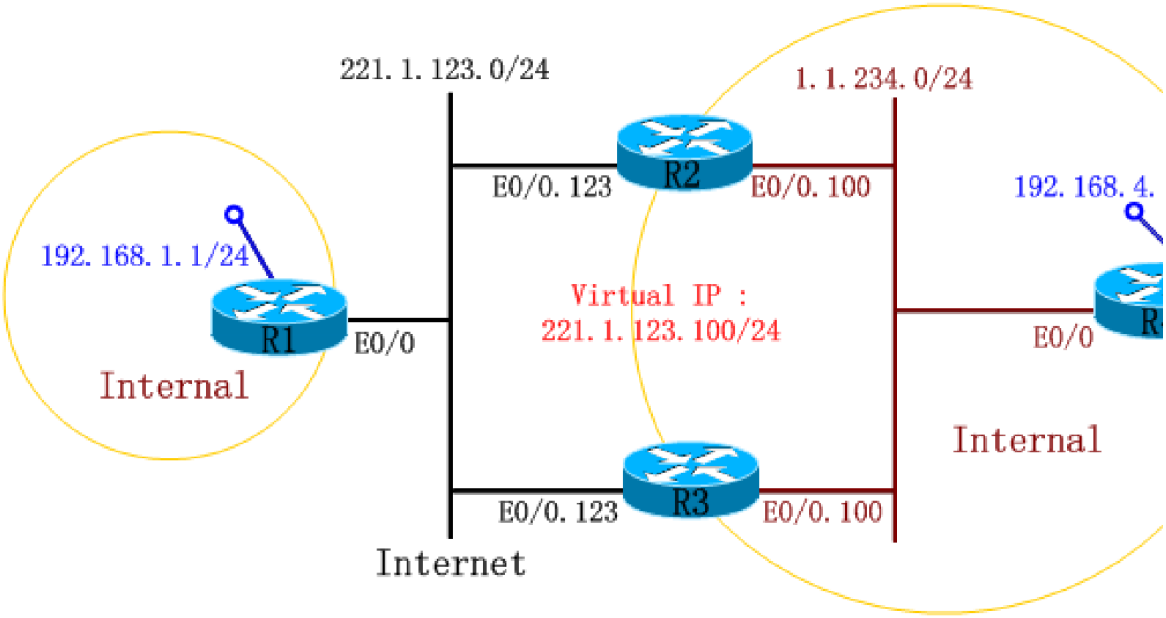
IPSec Phase II Policy:

```
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(cfg-crypto-trans)#mode tunnel
R2(config)#access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 123.1.1.1      <===先和第一个建立
R2(config-crypto-map)#set peer 123.1.1.3        <====当第一个断掉后,再和第二个建立
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address 101
...
```

Apply VPN Configuration

```
R2(config)#interface ethernet 0/0
R2(config-if)#crypto map huawei
-----------------------------------
R3(config)#interface ethernet 0/0
R3(config-if)#crypto map huawei
```

Redundancy VPN



HSRP:

```
R2(config)#int e0/0.123
R2(config-subif)#standby 1 ip 221.1.123.100
                         组名      virtual IP
R2(config-subif)#standby 1 preempt          <====允许抢占
R2(config-subif)#standby 1 priority 105      <====设置优先级(默认为100)
R2(config-subif)#standby 1 name HSRP      <====起一个名字(Redundancy VPN需要用到)
R2(config-subif)#standby 1 track e0/0.100       <===当R2的"e0/0.100"接口down掉后,立刻切换到standby路由器
```

```
R3(config)#int e0/0.123
R3(config-subif)#standby 1 ip 221.1.123.100
R3(config-subif)#standby 1 preempt
R3(config-subif)#standby 1 name HSRP
R3(config-subif)#standby 1 track e0/0.100
```

```
R2#show standby
Ethernet0/0.123 - Group 1
  State is Active
    2 state changes, last state change 00:37:58
  Virtual IP address is 221.1.123.100
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.276 secs
  Preemption enabled
  Active router is local
```

```
  Standby router is 221.1.123.3, priority 100 (expires in 9.360 sec)
  Priority 105 (configured 105)
    Track interface Ethernet0/0.100 state Up decrement 10
  IP redundancy name is "HSRP" (cfgd)

R1#ping 221.1.123.100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 221.1.123.100, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/4/4 ms
--------------------------------------------------------------------------------
-----
```

配置路由:

```
R1(config)#ip route 192.168.4.0 255.255.255.0 221.1.123.100     <====也可用反向路由注入
----------------------------
R2(config)#router ospf 110
R2(config-router)#net 1.1.234.0 0.0.0.255 area 0
-------------------------------------------------
R3(config)#router ospf 110
R3(config-router)#net 1.1.234.0 0.0.0.255 area 0
-------------------------------------------------
R4(config)#router ospf 110
R4(config-router)#net 1.1.234.0 0.0.0.255 area 0
R4(config-router)#net 192.168.4.0 0.0.0.255 area 0
```

IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 1
R2(config-isakmp)#authentication pre-share
R2(config)#crypto isakmp key wolf address 221.1.123.1
```

IPSec Phase II Policy:

```
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(config)#crypto map huawei 10 ipsec-isakmp   <===在"show run"时出现"!Incomplete",就是没有配置完
R2(config-crypto-map)#set peer 221.1.123.1
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#match address VPN
R2(config-crypto-map)#reverse-route      <====反向路由注入
R2(config)#ip access-list extended VPN
R2(config-ext-nacl)#permit ip 192.168.4.0 0.0.0.255 192.168.1.0 0.0.0.255
```

接口调用:

```
R2(config)#int e0/0.123
```

R2(config-subif)#crypto map huawei redundancy HSRP　　<===在此命令应用到接口后,查路由时:会出现一条静态路由("S　　192.168.1.0/24 [1/0] via 221.1.123.1")

以上R3的配置和R2类同　<===但R3没有一条静态路由,因为R3为"standby"

重分布:

```
R2(config)#router ospf 110
R2(config-router)#redistribute static subnets route-map s2o
R2(config)#route-map s2o
R2(config-route-map)#match ip address s2o
R2(config)#ip access-list standard s2o
R2(config-std-nacl)#permit 192.168.1.0
----------------------------------------
R3(config)#router ospf 110
R3(config-router)#redistribute static subnets route-map s2o
R3(config)#route-map s2o
R3(config-route-map)#match ip address s2o
R3(config)#ip access-list standard s2o
R3(config-std-nacl)#permit 192.168.1.0
```

R2(config)#crypto isakmp keepalive 20　　<===发送DPD帧,检测

```
-------------------------------------
R3(config)#crypto isakmp keepalive 20
-------------------------------------
R1(config)#crypto isakmp keepalive 20

R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config)#crypto isakmp key wolf address 221.1.123.100
R1(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R1(config)#crypto map huawei 10 ipsec-isakmp
R1(config-crypto-map)#set peer 221.1.123.100
R1(config-crypto-map)#set transform-set cisco
R1(config-crypto-map)#match address VPN
R1(config)#ip access-list extended VPN
R1(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255
R1(config)#int e0/0
R1(config-if)#crypto map huawei
```

反向路由注入:

1. Lan to Lan RRI 规则:
   以感兴趣流的目的为网络,以 PEER 地址为下一跳,注入这么一条静态路由

2. Soft Client / EZVPN client mode
   Server端的RRI规则:
   以分配给Client 的地址为网络(32位),以这个client的公网IP为下一跳,注入一条32位主机路由

3. EZVPN Hardware network-extension mode
   Server 端 RRI 注入规则:
   以Client端的内部网络为网络,以这个client的公网IP为下一跳,注入一条静态路由

4. Remote VPN 的 RRI 跟 Lan-to-Lan 不同:
   Remote VPN 是当client 拨入VPN 时注入的
   Lan-to-Lan 是配置完就马上注入的

问题1: 一个有crypto map 的接口,收到一个明文的数据包?
答: 收到明文 如果满足接口map的 感兴趣流 drop了,不满足 就当没做vpn

问题2: 一个没有crypto map 的接口,收到一个ESP的数据包?
答: 他回去查看他的SA数据库,然后就解密了.

IPSEC 穿越 PAT

IPSEC穿越PAT(应用层网关)

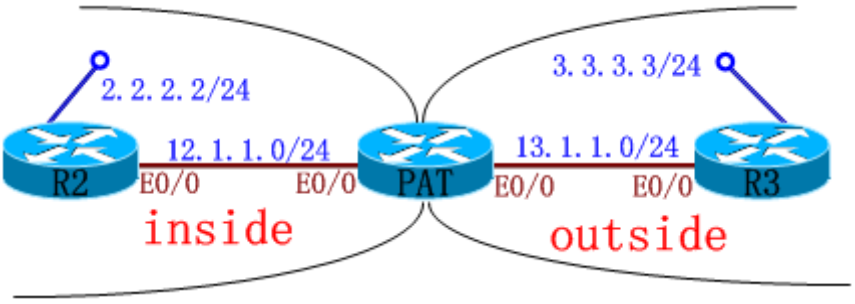
技术特点:

正常的IPSEC VPN 是不能穿越PAT的,CISCO的IPSEC VPN为了穿越PAT提供了很多扩展的功能.

例如: Nat - t , ipsec over udp ,ipsec over tcp , 应用层网关.

我们这个试验主要要演示应用层网关的功能,它是NAT设备提供的一种扩展功能.其他功能会在advance vpn部分讲解.

实验1:




```
PAT(config)#ip access-list extended pat        <====="pat"是一个名字
PAT(config-ext-nacl)#permit ip 12.1.1.0 0.0.0.255 any
PAT(config)#int e0/0.12
PAT(config-subif)#ip nat inside
PAT(config)#int e0/0.13
PAT(config-subif)#ip nat outside
PAT(config)#ip nat inside source list pat interface ethernet 0/0.13
overload  <===一段地址(12.1.1.0 0.0.0.255 any)PAT成一个接口
```

IKE Phase I Policy:

```
R2(config)#crypto isakmp policy 3
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#group 2
R2(config)#crypto isakmp key 0 wolf address 13.1.1.3
-----------------------------------------------------
R3(config)#crypto isakmp policy 3
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#hash md5
```

```
R3(config-isakmp)#encryption 3des
R3(config-isakmp)#group 2
R3(config)#crypto isakmp key 0 wolf address 13.1.1.1       <===PAT设备"outside"接口
```
问: 为什么"R3"不能直接PING通"R2"呢??
答: 由于"13.1.1.1"是PAT设备的接口,但PAT设备又没有与"R3"建立VPN.

IPSec Phase II Policy:
```
R2(config)#crypto ipsec nat-transparency udp-encapsulation     <===此命令为:打开"NAT-T".在26系列(版本:12.2)及以上路由器默认启用,"show run"也看不到
R2(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R2(cfg-crypto-trans)#mode tunnel
R2(config)#access-list 101 permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255
R2(config)#crypto map huawei 10 ipsec-isakmp
R2(config-crypto-map)#set peer 13.1.1.3
R2(config-crypto-map)#set transform-set cisco
R2(config-crypto-map)#set pfs
R2(config-crypto-map)#match address 101
-----------------------------------------------
R3(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac
R3(cfg-crypto-trans)#mode tunnel
R3(config)#access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
R3(config)#crypto map huawei 10 ipsec-isakmp
R3(config-crypto-map)#set peer 13.1.1.1           <===PAT设备"outside"接口
R3(config-crypto-map)#set transform-set cisco
R3(config-crypto-map)#set pfs
R3(config-crypto-map)#match address 101
```

Apply VPN Configuration
```
R2(config)#interface ethernet 0/0
R2(config-if)#crypto map huawei
-----------------------------------
R3(config)#interface ethernet 0/0
R3(config-if)#crypto map huawei
```

调试:
```
R2#ping 3.3.3.3 source loopback 0      <===只有"inside"发起连接,才能建立IPSEC道
R3#ping 2.2.2.2 source loopback 0          即只有"R2"先PING"R3",建立IPSEC道,"R3"才能PING通"R2"

PAT#show ip nat translations        <===转换项
Pro Inside global      Inside local       Outside local      Outside global
```

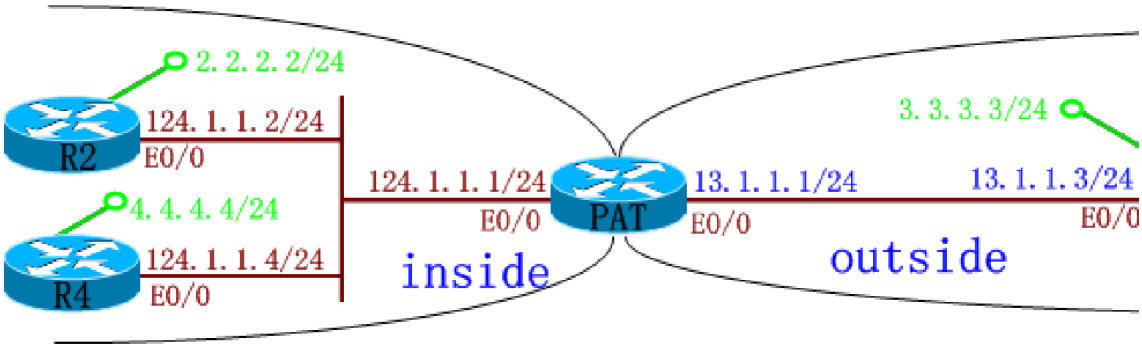
```
udp 13.1.1.1:4500       12.1.1.2:4500       13.1.1.3:4500
13.1.1.3:4500
------------------------------------------------------------------------------
----
R2#show cry ipsec sa

interface: Ethernet0/0
    Crypto map tag: huawei, local addr. 12.1.1.2

   protected vrf:
   local  ident (addr/mask/prot/port): (2.2.2.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (3.3.3.0/255.255.255.0/0/0)
   current_peer: 13.1.1.3:4500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 106, #pkts encrypt: 106, #pkts digest 106
    #pkts decaps: 106, #pkts decrypt: 106, #pkts verify 106
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 12.1.1.2, remote crypto endpt.: 13.1.1.3
     path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
     current outbound spi: 56ABE0AE

     inbound esp sas:
      spi: 0x71A6F86C(1906767980)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel UDP-Encaps, }    <===标准的NAT-T封装
        slot: 0, conn id: 2000, flow_id: 1, crypto map: huawei
        sa timing: remaining key lifetime (k/sec): (4491694/1405)
        IV size: 8 bytes
        replay detection support: Y

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x56ABE0AE(1454104750)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel UDP-Encaps, }
        slot: 0, conn id: 2001, flow_id: 2, crypto map: huawei
        sa timing: remaining key lifetime (k/sec): (4491694/1404)
```

IV size: 8 bytes
replay detection support: Y

outbound ah sas:

outbound pcp sas:

实验2:　　穿越PAT时,包结构:　...| UDP | ESP | DATA |...



配置:

```
......
R3(config)#access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
R3(config)#access-list 101 permit ip 3.3.3.0 0.0.0.255 4.4.4.0 0.0.0.255
......

R3#show access-lists 101
Extended IP access list 101
    10 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
    20 permit ip 3.3.3.0 0.0.0.255 4.4.4.0 0.0.0.255

PAT#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside
global
udp 13.1.1.1:1024      124.1.1.4:4500     13.1.1.3:4500
13.1.1.3:4500
udp 13.1.1.1:4500      124.1.1.2:4500     13.1.1.3:4500
13.1.1.3:4500
```

端口号:第一个包过来端口号:为"4500"(规定); 第二个包过来端口号:为"1024"(随机产生)

```
udp 13.1.1.1:500       124.1.1.4:500      13.1.1.3:500       13.1.1.3:500
```

端口号为:"500",是IKE的协商包.

```
----------------------------------------
R3#show crypto engine connections active
  ID Interface            IP-Address      State  Algorithm
```

```
Encrypt  Decrypt
   1 Ethernet0/0          13.1.1.3        set    HMAC_MD5+3DES_56_C
0        0
2000 Ethernet0/0          13.1.1.3        set    HMAC_SHA+DES_56_CB
0      114
2001 Ethernet0/0          13.1.1.3        set    HMAC_SHA+DES_56_CB
114      0
2002 Ethernet0/0          13.1.1.3        set    HMAC_SHA+DES_56_CB
0      104
2003 Ethernet0/0          13.1.1.3        set    HMAC_SHA+DES_56_CB
104      0
```

假设输入以下命令,

```
R2(config)#no crypto ipsec nat-transparency  udp-encapsulation    <===关闭"NAT-T",只剩下"ESP"
R2#clear crypto isakmp
R2#clear crypto sa
R3(config)#no crypto ipsec nat-transparency udp-encapsulation
R3#clear crypto isakmp
R3#clear crypto sa
R4(config)#no crypto ipsec nat-transparency udp-encapsulation
R4#clear crypto isakmp
R4#clear crypto sa
```

原来:

```
R2#show crypto ipsec sa
inbound esp sas:
      spi: 0x6BA32953(1805855059)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel UDP-Encaps, }   <===包结构:   ...| UDP | ESP | DATA |...
        slot: 0, conn id: 2000, flow_id: 1, crypto map: huawei
        sa timing: remaining key lifetime (k/sec): (4547307/3587)
        IV size: 8 bytes
        replay detection support: Y
```

现在:

```
R2#show crypto ipsec sa
     inbound esp sas:
      spi: 0x7E919BB2(2123471794)
        transform: esp-des esp-sha-hmac ,
        in use settings ={Tunnel, }       <====没有"UPD"封装,只剩下"ESP"了
        slot: 0, conn id: 2000, flow_id: 1, crypto map: huawei
        sa timing: remaining key lifetime (k/sec): (4477757/3134)
```

IV size: 8 bytes
replay detection support: Y

问: 为什么在不是"NAT-T"的情况下,一样可以PING通?
答: PAT设备是根据 "ESP"的"SPI"号,去区分哪个"SPI"号属于哪个IP地址.
PAT(config)#crypto ipsec nat-transparency spi-matching <===打开"SPI"功能
PAT#clear ip nat translation * <===清除转换项

PAT#show ip nat translations
Pro Inside global Inside local Outside local Outside global
esp 13.1.1.1:0 124.1.1.2:0 13.1.1.3:0
13.1.1.3:29E28D9E
esp 13.1.1.1:0 124.1.1.2:85043DDB 13.1.1.3:0 13.1.1.3:0
"ESP"的"SPI"号
esp 13.1.1.1:0 124.1.1.4:27D433F 13.1.1.3:0 13.1.1.3:0
icmp 13.1.1.1:5 124.1.1.2:5 13.1.1.3:5 13.1.1.3:5
"ICMP"的"SPI"号("inside"的路由器去PING"outside"的Peer地址)
icmp 13.1.1.1:9 124.1.1.4:9 13.1.1.3:9 13.1.1.3:9
udp 13.1.1.1:2 124.1.1.2:500 13.1.1.3:500 13.1.1.3:500
esp 13.1.1.1:0 124.1.1.4:0 13.1.1.3:0
13.1.1.3:AD7912D1
udp 13.1.1.1:4500 124.1.1.2:4500 13.1.1.3:4500
13.1.1.3:4500
udp 13.1.1.1:500 124.1.1.4:500 13.1.1.3:500 13.1.1.3:500

IPSEC穿越PAT(应用层网关)

1. 静态转换 ---- 在PAT设备上,每一个接口/子接口(outside接口),只能对应一个"inside"的IP地址.

从内部发起:

PAT(config)#ip nat inside source static esp 124.1.1.2 interface ethernet 0/0.13 <==根据实验2的图
内网的路由器接口地址 PAT设备的"outside"接口

PAT#show ip nat translations
Pro Inside global Inside local Outside local Outside global
esp 13.1.1.1:0 124.1.1.2:0 13.1.1.3:0 13.1.1.3:0
udp 13.1.1.1:5 124.1.1.4:500 13.1.1.3:500 13.1.1.3:500
udp 13.1.1.1:6 124.1.1.4:500 13.1.1.3:500 13.1.1.3:500
esp 13.1.1.1:0 124.1.1.2:0 --- ---

udp 13.1.1.1:500 124.1.1.2:500 13.1.1.3:500 13.1.1.3:500
udp 13.1.1.1:500 124.1.1.2:500 13.1.1.3:500 13.1.1.3:500

从外部发起:

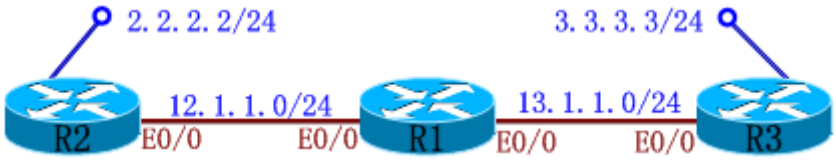
内网的路由器接口地址的IKE协商包端口号

PAT(config)#ip nat inside source static udp 124.1.1.2 500 interface ethernet 0/0.13 500

PAT设备的"outside"接口的IKE协商包端口号

PAT#show run | in static
ip nat inside source static udp 124.1.1.2 500 interface Ethernet0/0.13 500
ip nat inside source static esp 124.1.1.2 interface Ethernet0/0.13

ACL
实验3:



在R1的左边的端口上配置一条ACL:
R1(config)#ip access-list extended ACLR1IN
R1(config-ext-nacl)#permit udp host 12.1.1.2 host 13.1.1.3 eq 500 <===放行协商包
R1(config-ext-nacl)#permit esp host 12.1.1.2 host 13.1.1.3 <===放行数据流
R1(config)#int e0/0.12
R1(config-subif)#ip access-group ACLR1IN in

在R2的右边的端口上配置一条ACL:
R2(config)#ip access-list extended ACLR2IN
R2(config-ext-nacl)#permit udp host 13.1.1.3 host 12.1.1.2 eq 500
R2(config-ext-nacl)#permit esp host 13.1.1.3 host 12.1.1.2
R2(config-ext-nacl)#deny ip any any log <===加上"log",会弹出LOG信息(不一定要加在这里,只要是permit/deny语句都可以加)
R2(config-ext-nacl)#int e0/0
R2(config-if)#ip access-group ACLR2IN in

在此前是PING不通的,因为两次检查访问列表(解密前/后各检查一次):
显示:
R2#
*Mar 1 01:01:49.411: %SEC-6-IPACCESSLOGDP: list ACLR2IN denied icmp 3.3.3.3 -> 2.2.2.2 (8/0), 1 packet

还需要配置上以下命令:

```
R2(config)#ip access-list extended ACLR2IN
R2(config-ext-nacl)#5 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
```

<===解密后的数据还要再检验一次

```
R2#show ip access-lists
Extended IP access list 101
    10 permit ip 2.2.2.0 0.0.0.255 3.3.3.0 0.0.0.255 (93 matches)
Extended IP access list ACLR2IN
    5 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255 (12 matches)
    40 permit udp host 13.1.1.3 host 12.1.1.2 eq isakmp (15 matches)
    50 permit esp host 13.1.1.3 host 12.1.1.2 (4 matches)
    100 deny ip any any
```

如上图:PAT设备,在R3上需要放行"UDP"的"4500"号端口

DMVPN

DMVPN----Dynamic Multipoint VPN

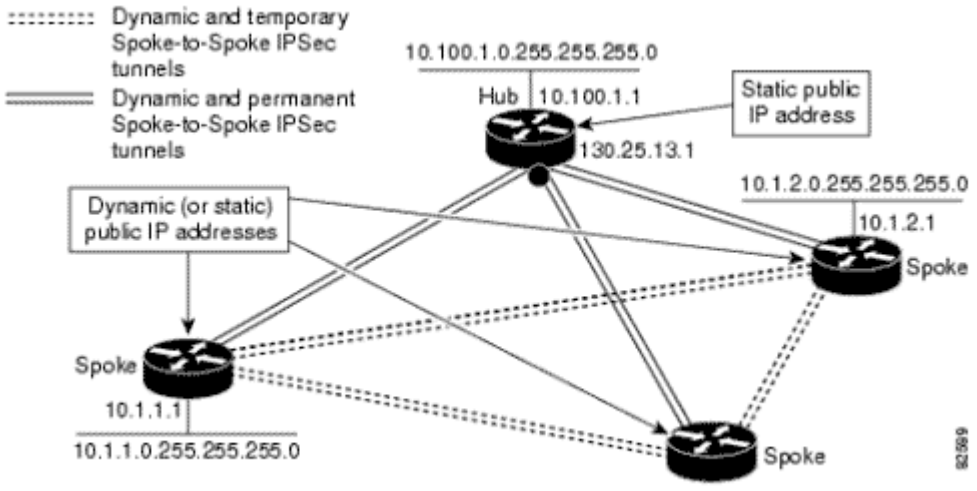
动态多点VPN（DMVPN）-顾名思义：DMVPN应用在多点到多点的复杂VPN网络环境中。

DMVPN的动态连接关系是通过hub-spokes模式来实现，它可以在两个或多个DMVPN成员的各自子网间动态建立基于GRE over ipsec连接的路由路径。根据目的前缀和下一跳结合路由协议来推断调用的GRE隧道和相关ipsec 保护策略。

DMVPN还要结合mGRE(Multipoint GRE interfaces)和NHRP（Next Hop Resolution Protocol）等相关知识。

大致流程：

1. 首先通过mGRE的封装代替p-t-p GRE封装来减少手动tunnel 数量及有效结合NHRP。
2. NHRP结合动态路由协议来支持mGRE的下一跳动态解析功能，以便根据路由协议动态建立保护网络间的临时tunnel.(保护网络要路由通告出去)
3. 在mGRE下采用ipsec profile 实现 ipsec automatic proxy 功能；保护gre封装流量，加密根据NHRP和动态路由协议建立起来的保护网络间的临时tunnel。（由于是动态方式，IKE isakmp remote peers 为 0.0.0.0）



配置案例

Hub Configuration for DMVPN

```
crypto isakmp policy 1
authentication pre-share
crypto isakmp key ikecisco address 0.0.0.0
!
crypto ipsec transform-set myset esp-des esp-md5-hmac
mode transport
```

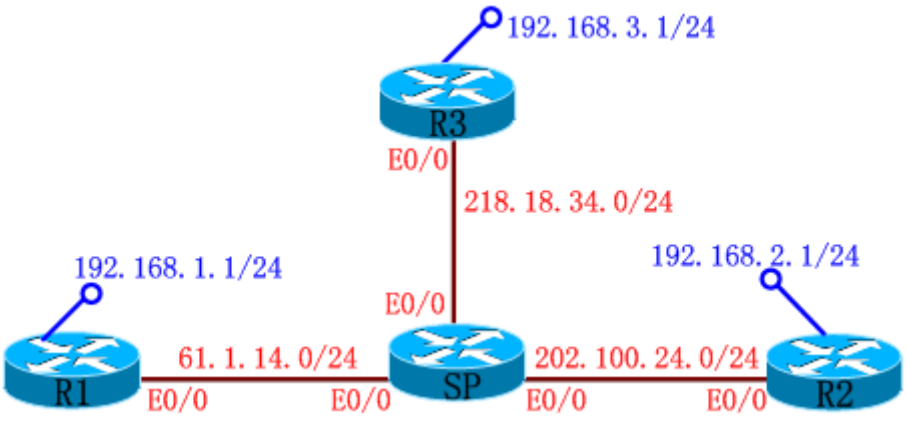
```
!
crypto ipsec profile vpnprof
set transform-set myset
!
interface Tunnel0
bandwidth 1000
ip address 10.0.0.1 255.255.255.0
ip mtu 1416（确保大包在加密之前分片，目的设备正常重组）
ip nhrp authentication nhrp-pwd（nhrp认证，在匹配后，调用mGRE属性）
ip nhrp map multicast dynamic（启用NHRP自动加入分支路由器到多播NHRP映射组中）
ip nhrp network-id 99（启用NHRP,在匹配后，调用mGRE属性）
ip nhrp holdtime 300
no ip split-horizon eigrp 1（当使用EIGRP协议时，屏蔽水平分割）
no ip next-hop-self eigrp 1 (当使用EIGRP时，直接建立动态SPOKE-TO-SPOKE隧道)
delay 1000
tunnel source Ethernet0
tunnel mode gre multipoint（设置隧道接口的封装模式为mGRE）
tunnel key cisco (tunnel 认证)
tunnel protection ipsec profile vpnprof(为隧道接口指定IPSEC模板)
!
interface Ethernet0
ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
network 10.0.0.0 0.0.0.255 area 0
network 192.168.0.0 0.0.0.255 area 0
```

Spoke Configuration for DMVPN

```
crypto isakmp policy 1
authentication pre-share
crypto isakmp key ikecisco address 0.0.0.0
!
crypto ipsec transform-set myset esp-des esp-md5-hmac
mode transport
!
crypto ipsec profile vpnprof
set transform-set myset
!
```

```
interface Tunnel0
bandwidth 1000
ip address 10.0.0.2 255.255.255.0
ip mtu 1416
ip nhrp authentication nhrp-pwd
ip nhrp map 10.0.0.1 172.17.0.1（为NHRP server hub 隧道地址10.0.0.1做物理地址绑定）
ip nhrp map multicast 172.17.0.1（在分支和HUB之间启动动态路由协议，并发送多播包到HUB路由器上）
ip nhrp network-of 99
ip nhrp holdtime 300
ip nhrp nhs 10.0.0.1（配置HUB路由器作为NHRP的下一跳服务器）
delay 1000
tunnel source Ethernet0
tunnel mode gre multipoint
tunnel key cisco
tunnel protection ipsec profile vpnprof
!
interface Ethernet0
ip address 172.16.0.2 255.255.255.0
!
interface Ethernet1
ip address 192.168.1.1 255.255.255.0
!
router eigrp 1
network 10.0.0.0 0.0.0.255
network 192.168.1.0 0.0.0.255
```

实验1:



第一步：建立 Dynamic Multipoint

```
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#ip address 172.16.1.3 255.255.255.0
HUB-R3(config-if)#tunnel source ethernet 0/0
HUB-R3(config-if)#tunnel mode gre multipoint      <===点到多点(不需要配
```

置:"tunnel"的"destination")
对比: HUB-R3(config-if)#tunnel mode gre ip <===点到点,默认配置(需要配置:"tunnel"的"destination")
HUB-R3(config-if)#tunnel key 12345   <===在多点中:"key"是必配(在点到点中:"tunnel"的"key"可不配)

SPOKE-R1(config)#interface tunnel 0
SPOKE-R1(config-if)#ip address 172.16.1.1 255.255.255.0
SPOKE-R1(config-if)#tunnel source ethernet 0/0
SPOKE-R1(config-if)#tunnel mode gre multipoint
SPOKE-R1(config-if)#tunnel key 12345

NHRP配置:
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#ip nhrp network-id 100   <====="network-id"号,可任意(但:HUB/SPOKE端要在同一个"network-id"里面)

SPOKE-R1(config)#interface tunnel 0
SPOKE-R1(config-if)#ip nhrp network-id 100
SPOKE-R1(config-if)#ip nhrp nhs 172.16.1.3   <===HUB的"tunnel"地址(nhs: nh server)
SPOKE-R1(config-if)#ip nhrp map 172.16.1.3 218.18.34.3   <==="tunnel"地址与公网地址的映射
                                "tunnel"地址 公网地址

HUB-R3#show ip nhrp   <====动态映射
172.16.1.1/32 via 172.16.1.1, Tunnel0 created 00:06:11, expire 01:53:48
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 61.1.14.1

SPOKE-R1#show ip nhrp   <====静态映射
172.16.1.3/32 via 172.16.1.3, Tunnel0 created 00:04:21, never expire
  Type: static, Flags: authoritative used
  NBMA address: 218.18.34.3   <===当"SPOKE"端的路由器重启后,就会到"SERVER"端的路由器上去注册,所以SERVER端的路由器的IP地址一定要是静态的IP地址.

SPOKE-R1#show ip nhrp nhs   <===查映射(只能在"SPOKE"端使用)
Legend:
  E=Expecting replies
  R=Responding
Tunnel0:
  172.16.1.3   RE   <===有响应

宣告:

```
SPOKE-R1(config)#router eigrp 100
SPOKE-R1(config-router)#no auto-summary
SPOKE-R1(config-router)#network 192.168.1.0 0.0.0.255
SPOKE-R1(config-router)#network 172.16.1.0 0.0.0.255
```

```
HUB-R3(config-if)#router eigrp 100
HUB-R3(config-router)#no auto-summary
HUB-R3(config-router)#net 192.168.3.0 0.0.0.255
HUB-R3(config-router)#net 172.16.1.0 0.0.0.255
```

发组播包:

```
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#ip nhrp map multicast dynamic    <===动态地发组播包(HUB)
SPOKE-R1(config)#interface tunnel 0
SPOKE-R1(config-if)#ip nhrp map multicast 218.18.34.3     <====静态地发组播包(SPOKE)
```

出现问题:"HUB-R3"有neighbor,但"SPOKE-R1"没有neighbor ??

```
HUB-R3(config-if)#
*Mar  1 00:44:49.071: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor
172.16.1.1 (Tunnel0) is up: new adjacency
HUB-R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q
Seq
                                            (sec)         (ms)       Cnt
Num
0   172.16.1.1              Tu0               14 00:00:38     1  5000  1
0
SPOKE-R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 100
```

解决: 1. 先将"HUB"和"SPOKE"端的TUNNEL接口"shut down"

2. 然后"no shut down"HUB端的接口
3. 再"no shut down"SPOKE端的接口

```
SPOKE-R1#show ip route eigrp
D     192.168.3.0/24 [90/297372416] via 172.16.1.3, 00:00:19, Tunnel0
```

关闭HUB的"tunnel"接口的水平分割:
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#no ip split-horizon eigrp 100　　<===关闭EIGRP的"tunnel"接口的水平分割
或者　　HUB-R3(config-if)#no ip split-horizon　　<===这是关闭RIP的水平分割

出现问题: 在"SPOKE-R1"路由器上去往"SPOKE-R2"路由器的数据包要经过"HUB-R3"路由器 ??
SPOKE-R1#show ip route eigrp　　下一跳
D　　192.168.2.0/24 [90/310172416] via 172.16.1.3, 00:31:09, Tunnel0
<===在"SPOKE-R1"路由器上去往"SPOKE-R2"路由器的数据包要经过"HUB-R3"路由器
D　　192.168.3.0/24 [90/297372416] via 172.16.1.3, 00:31:09, Tunnel0
解决:
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#no ip next-hop-self eigrp 100　　<===只有12.3以上版本的路由器才有此命令

SPOKE-R2#show ip route eigrp
D　　192.168.1.0/24 [90/310172416] via 172.16.1.1, 00:03:46, Tunnel0
D　　192.168.3.0/24 [90/297372416] via 172.16.1.3, 00:03:47, Tunnel0
SPOKE-R2#show ip nhrp
172.16.1.1/32 via 172.16.1.1, Tunnel0 created 00:00:17, expire 01:47:40
<===PING了之后才会出现
  Type: dynamic, Flags: router
  NBMA address: 61.1.14.1
172.16.1.3/32 via 172.16.1.3, Tunnel0 created 00:52:16, never expire
  Type: static, Flags: authoritative used
  NBMA address: 218.18.34.3

第二步: 建立 VPN

IKE Phase I Policy:
HUB-R3(config)#crypto isakmp policy 1
HUB-R3(config-isakmp)#authentication pre-share
HUB-R3(config)#crypto isakmp key 0 wolf address 0.0.0.0 0.0.0.0

IPSec Phase II Policy:
HUB-R3(config)#crypto ipsec transform-set cisco esp-des esp-sha-hmac

```
HUB-R3(cfg-crypto-trans)#mode transport
HUB-R3(config)#crypto ipsec profile huawei
HUB-R3(config-crypto-map)#set transform-set cisco
```

接口调用:

```
HUB-R3(config)#interface tunnel 0
HUB-R3(config-if)#tunnel protection ipsec profile huawei
```

更改MTU值:

```
SPOKE-R1(config)#int tunnel 0
SPOKE-R1(config-if)#ip mtu 1436      <===更改"tunnel"的MTU值(MTU大于1436将会分片),CISCO建议为:1400
```

Easy VPN

Easy VPN

1. 简化client端的配置
2. 对client端进行集中化管理,向client端推送一些策略.

3. client端有一些预配置(IKE Policy & IPSEC transform sets)
4. 由client端发起连接(IKE Phase 1 process).
5. 不支持group 1,支持group 2；不支持AH,支持ESP；支持Tunnel mode ,不支持Transport mode
6. 使用Group Policy只能使用RADIUS服务器

Easy VPN server:Router/PIX/VPN
Easy VPN remote:Router/PIX/VPN(3002)/Software client

实验 1 ：

环境:

```
Internet(config)#line vty 0 4
Internet(config-line)#no login          <===不使用登录密码
Internet(config-line)#privilege level 15  <====将权限提到第15级(telnet进去时,直接进入特权模式)

PAT(config)#ip access-list extended P
PAT(config-ext-nacl)#permit ip 218.18.100.0 0.0.0.255 any
PAT(config)#int e0/0.218
PAT(config-subif)#ip nat inside
PAT(config)#int e0/0.100
PAT(config-subif)#ip nat outside
PAT(config)#ip nat inside source list P interface ethernet 0/0.100 overload
```

PC机:

```
  Internet#show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 01:01:44
* 66 vty 0                 idle                 00:00:00   100.1.1.1
-------------------------------------------------------------------------

PAT#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 100.1.1.1:1024    218.18.100.241:1024 100.1.1.3:1024
```

100.1.1.3:1024
tcp 100.1.1.1:2501 218.18.100.241:2501 100.1.1.3:23 100.1.1.3:23

VpnServer(config)#username ciscovpn password ciscovpn
以上的用户名和密码应用于以下图中:

VpnServer(config)#aaa new-model <====起用AAA
VpnServer(config)#aaa authentication login NOAU none <===定义"NOAU"
VpnServer(config)#line con 0
VpnServer(config-line)#login authentication NOAU <===调用"NOAU"
VpnServer(config)#line vty 0 4
VpnServer(config-line)#login authentication NOAU
VpnServer(config)#aaa authentication login xauth local <===login的名字"xauth",本地
VpnServer(config)#aaa authorization network modeconf local <===网络的授权名字:"modeconf",本地

IKE Phase I Policy:
VpnServer(config)#crypto isakmp policy 10
VpnServer(config-isakmp)#authentication pre-share
VpnServer(config-isakmp)#group 2 <==="group"必须为"2"(在EasyVPN中)
VpnServer(config-isakmp)#hash md5 <===软件作为client,必须为"MD5"; 硬件作为client,则没规定
VpnServer(config)#crypto isakmp client configuration group EZVPN <===相当于"pre-share"KEY的名字
VpnServe(config-isakmp-group)#key cisco123 <===基于证书时不需要KEY <==相当于"pre-share"KEY的密码
以上两条命令的"group"和"key"用于以下图中:

VpnServe(config-isakmp-group)#pool ezvpnpool <===地址池的名字

VpnServe(config-isakmp-group)#acl 101 <===如不调用此列表,将是Tunnel everything; 如调用此列表,将是split Tunnel(即: 定义了哪种流量, 哪种流量才加密)

```
VpnServe(config-isakmp-group)#?
ISAKMP group policy config commands:
  access-restrict  Restrict clients in this group to an interface
<==限制client从哪一个接口拨入VPN服务器
  acl              Specify split tunneling inclusion access-list number
  dns              Specify DNS Addresses     <===给client端推送一个DNS的
地址
  domain           Set default domain name to send to client
  exit             Exit from ISAKMP client group policy configuration
mode
  group-lock       Enforce group lock feature  <===用户和组的绑定(即:这个
用户必须属于这个组才能用)
  key              pre-shared key/IKE password
  no               Negate a command or set its defaults
  pool             Set name of address pool
  wins             Specify WINS Addresses
```

```
VpnServer(config)#ip local pool ezvpnpool 192.168.168.100 192.168.168.200
<===定义地址池(分配地址给client端)
VpnServer(config)#access-list 101 permit ip 192.168.4.0 0.0.0.255 any
VpnServer(config)#crypto isakmp profile ISA
VpnServer(conf-isa-prof)#match identity group EZVPN
VpnServer(conf-isa-prof)#client authentication list xauth
VpnServer(conf-isa-prof)#isakmp authorization list modeconf
地址的推送方式:
VpnServer(conf-isa-prof)#client configuration address respond   <===cisco
的client软件版本为3.0以上使用(respond/pull),客户端去询问服务器端要地址.
或者: VpnServer(conf-isa-prof)#client configuration address initiate
<===cisco的client软件版本为3.0以下使用(push),服务器端向客户端推送地址.

IPSec Phase II Policy:
VpnServer(config)#crypto ipsec transform-set MYSET esp-des esp-md5-hmac
VpnServer(config)#crypto dynamic-map DMAP 10
VpnServer(config-crypto-map)#set transform-set MYSET
VpnServer(config-crypto-map)#set isakmp-profile ISA
VpnServer(config-crypto-map)#reverse-route
VpnServer(config)#crypto map SMAP 10 ipsec-isakmp dynamic DMAP

Apply VPN Configuration:
VpnServer(config)#interface e0/0
```

```
VpnServer(config-if)#crypto map SMAP

------------------------------------------------------
VpnServer#show ip local pool

 Pool                    Begin           End             Free  In use
 ezvpnpool               192.168.168.100 192.168.168.200  100       1
------------------------------------------------------
VpnServer#show ip local pool ezvpnpool
 Pool                    Begin           End             Free  In use
 ezvpnpool               192.168.168.100 192.168.168.200  100       1
Available addresses:
   192.168.168.105
   192.168.168.106
   ......
   192.168.168.100    IKE Addr IDB
   192.168.168.101    IKE Addr IDB
   192.168.168.102    IKE Addr IDB
   192.168.168.103    IKE Addr IDB
Inuse addresses:
   192.168.168.104    IKE Addr IDB
------------------------------------------------------
VpnServer#show ip route
     100.0.0.0/24 is subnetted, 1 subnets
C       100.1.1.0 is directly connected, Ethernet0/0
C    192.168.4.0/24 is directly connected, Loopback0
     192.168.168.0/32 is subnetted, 1 subnets
S       192.168.168.104 [1/0] via 100.1.1.1          <===反向路由注入第二条规则

硬件CLIENT端：
BranchCompany(config)#crypto ipsec client ezvpn cisco    <===名字
BranchCompany(config-crypto-ezvpn)#peer 100.1.1.4
BranchCompany(config-crypto-ezvpn)#group EZVPN key cisco123   <===group名字和密码
BranchCompany(config-crypto-ezvpn)#mode client
BranchCompany(config-crypto-ezvpn)#connect manual   <===可选自动/手动

Apply VPN Configuration:
BranchCompany(config)#int e0/0
BranchCompany(config-if)#crypto ipsec client ezvpn cisco outside   <===调用
BranchCompany(config)#int loopback 0
BranchCompany(config-if)#crypto ipsec client ezvpn cisco inside
```

拨号:

```
BranchCompany#crypto ipsec client ezvpn connect
出现提示:
  BranchCompany#
  *Mar  1 13:00:06.827: EZVPN(cisco): Pending XAuth Request, Please enter
the following command:
  *Mar  1 13:00:06.827: EZVPN: crypto ipsec client ezvpn xauth
BranchCompany#crypto ipsec client ezvpn xauth
Username: ciscovpn
Password: ciscovpn
------------------------------------------------
BranchCompany#show crypto ip client ezvpn    <===以下是拨不成功的
(Password: Disallowed)
Easy VPN Remote Phase: 4
Tunnel name : cisco
Inside interface list: Loopback0
Outside interface: Ethernet0/0
Current State: CONNECT_REQUIRED
Last Event: RESET
Save Password: Disallowed
Current EzVPN Peer: 100.1.1.4
------------------------------------------------
BranchCompany#show crypto ipsec client ezvpn    <===以下是拨成功的
Easy VPN Remote Phase: 4
Tunnel name : cisco
Inside interface list: Loopback0
Outside interface: Ethernet0/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 192.168.168.105     <===这是分配的IP地址
Mask: 255.255.255.255
Save Password: Disallowed
Split Tunnel List: 1         <===访问以下网络要加密
       Address    : 192.168.4.0
       Mask       : 255.255.255.0
       Protocol   : 0x0
       Source Port: 0
       Dest Port  : 0
Current EzVPN Peer: 100.1.1.4
------------------------------------------------

BranchCompany#telnet 192.168.4.1 /source-interface loopback 0  <===以
client端的内部网络去上网(client的内部网络必须要有去server端的内部网络的路
```

由)

```
BranchCompany#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside
global
tcp 100.1.1.2:43004    192.168.2.1:43004  100.1.1.3:23       100.1.1.3:23
tcp 192.168.168.108:55185 192.168.2.1:55185 192.168.4.1:23
192.168.4.1:23
```

```
BranchCompany#telnet 192.168.4.1 /source-interface loopback 0
Trying 192.168.4.1 ... Open
VpnServer>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:01:45
* 66 vty 0                 idle                 00:00:00 192.168.168.108
```

```
BranchCompany#show ip nat statistics
Total active translations: 1 (0 static, 1 dynamic; 1 extended)
Outside interfaces:
  Ethernet0/0
Inside interfaces:
  Loopback0
Hits: 533  Misses: 0
CEF Translated packets: 254, CEF Punted packets: 0
Expired translations: 7
Dynamic mappings:
-- Inside Source
[Id: 4] access-list internet-list interface Ethernet0/0 refcount 0
[Id: 3] access-list enterprise-list pool cisco refcount 1
 pool cisco: netmask 255.255.255.0
        start 192.168.168.108 end 192.168.168.108
        type generic, total addresses 1, allocated 1 (100%), misses 0
Queued Packets: 0
```

清除连接:

```
BranchCompany#clear crypto ipsec client ezvpn
```

<===清除EZVPN的连接(好像只能在CLIENT端做)

或者:　　BranchCompany#clear crypto sa
　　BranchCompany#clear crypto isakmp

网络扩展模式:　　<====server端不会再下发IP地址,更像一个LAN-TO-LAN

```
BranchCompany(config)#crypto ipsec client ezvpn cisco
BranchCompany(config-crypto-ezvpn)#mode network-extension      <===网络扩展模式

BranchCompany#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  Ethernet0/0
Inside interfaces:
  Loopback0
Hits: 699  Misses: 0
CEF Translated packets: 335, CEF Punted packets: 0
Expired translations: 8
Dynamic mappings:
-- Inside Source
[Id: 9] access-list internet-list interface Ethernet0/0 refcount 0
<===上面是PAT成两个,现在是PAT一个
Queued Packets: 0

BranchCompany#show ip nat translations        <===没有将"192.168.2.1"PAT成别的
Pro Inside global      Inside local       Outside local      Outside global
tcp 100.1.1.2:13702    192.168.2.1:13702  100.1.1.3:23       100.1.1.3:23

BranchCompany#telnet 192.168.4.1 /source-interface loopback 0
Trying 192.168.4.1 ... Open      <=====原地址(192.168.2.1)没更改,直接过去,更像是Lan-to-Lan
VpnServer>show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:21:05
* 66 vty 0                 idle                 00:00:00 192.168.2.1
<===这个地址没有更改
```

VPN 3000

VPN3000的默认密码:　用户名:admin　密码:admin

VPN3000的主菜单模式:　　<===在任何时候按"h",然后回车,将会返回主菜单模式

```
                Welcome to
               Cisco Systems
       VPN 3000 Concentrator Series
          Command Line Interface
Copyright (C) 1998-2005 Cisco Systems, Inc.
1) Configuration
2) Administration
3) Monitoring
4) Save changes to Config file
5) Help Information
6) Exit
VPN3000: Main ->
```

清空重启VPN3000: 2 --> 3 --> 2 --> 3 --> 2

```
Rebooting VPN 3000 Concentrator now.
Resetting System...
Boot-ROM Initializing...
Boot configured 32Mb of RAM.
...
Loading image ..........
Verifying image checksum ............
Active image loaded and verified...
Starting loaded image...
Starting power-up diagnostics...
...
pSH+ Copyright (c) Integrated Systems, Inc., 1992.
Cisco Systems, Inc./VPN 3000 Concentrator Version 4.7.2.A Aug 19 2005
01:03:59
Features:
Initializing VPN 3000 Concentrator ...
Waiting for CAPI initialization to complete...
Initialization Complete...Waiting for Network...
1 01/01/1999 00:00:05.740 SEV=1 EVENT/37 RPT=1
Reset Reason : 2 (Hardware-Reset)
```

快速配置模式:　　<===进入此模式,证明VPN3000是清空重启的

```
Welcome to
                Cisco Systems
        VPN 3000 Concentrator Series
           Command Line Interface
Copyright (C) 1998-2005 Cisco Systems, Inc.
 -- : Set the time on your device. The correct time is very important,
 -- : so that logging and accounting entries are accurate.
 -- : Enter the system time in the following format:
 -- :       HH:MM:SS.  Example  21:30:00  for 9:30 PM
> Time
Quick -> [ 00:06:02 ] 13:26:00
 -- : Enter the date in the following format.
 -- : MM/DD/YYYY  Example 06/12/1999  for June 12th 1999.
> Date
Quick -> [ 01/01/1999 ] 03/16/2007
 -- :  Set the time zone on your device. The correct time zone is very
 -- :  important so that logging and accounting entries are accurate.
 -- :  Enter the time zone using the hour offset from GMT:
 -- :  -12 : Kwajalein   -11 : Samoa      -10 : Hawaii         -9 :
Alaska
 -- :   -8 : PST          -7 : MST         -6 : CST            -5 : EST
```

```
 -- :   -4 : Atlantic     -3 : Brasilia  -3.5 : Newfoundland   -1 : Mid-
Atlantic
 -- :   -1 : Azores        0 : GMT         +1 : Paris           +2 : Cairo
 -- :   +3 : Kuwait     +3.5 : Tehran      +4 : Abu Dhabi     +4.5 : Kabul
 -- :   +5 : Karachi    +5.5 : Calcutta +5.75 : Kathmandu       +6 :
Almaty
 -- : +6.5 : Rangoon      +7 : Bangkok     +8 : Singapore       +9 : Tokyo
 -- : +9.5 : Adelaide    +10 : Sydney     +11 : Solomon Is.    +12 :
Marshall Is.
> Time Zone
Quick -> [ 8 ] 8
1) Enable Daylight Savings Time Support
2) Disable Daylight Savings Time Support
Quick -> [ 1 ]
This table shows current IP addresses.

  Intf         Status       IP Address/Subnet Mask          MAC Address
-------------------------------------------------------------------------
------
Ether1-Pri|Not Configured|        0.0.0.0/0.0.0.0        |
Ether2-Pub|Not Configured|        0.0.0.0/0.0.0.0        |
-------------------------------------------------------------------------
------
DNS Server(s): DNS Server Not Configured
DNS Domain Name:
Default Gateway: Default Gateway Not Configured
** An address is required for the private interface. **
> Enter IP Address
Quick Ethernet 1 -> [ 0.0.0.0 ] 10.1.1.100    <===必须要在快速模式下配置
Waiting for Network Initialization...
> Enter Subnet Mask
Quick Ethernet 1 -> [ 255.0.0.0 ] 255.255.255.0
1) Ethernet Speed 10 Mbps
2) Ethernet Speed 100 Mbps
3) Ethernet Speed 10/100 Mbps Auto Detect
Quick Ethernet 1 -> [ 3 ]
1) Enter Duplex - Half/Full/Auto
2) Enter Duplex - Full Duplex
3) Enter Duplex - Half Duplex
Quick Ethernet 1 -> [ 1 ]
> MTU (68 - 1500)
Quick Ethernet 1 -> [ 1500 ]
1) Modify Ethernet 1 IP Address (Private)
2) Modify Ethernet 2 IP Address (Public)
```

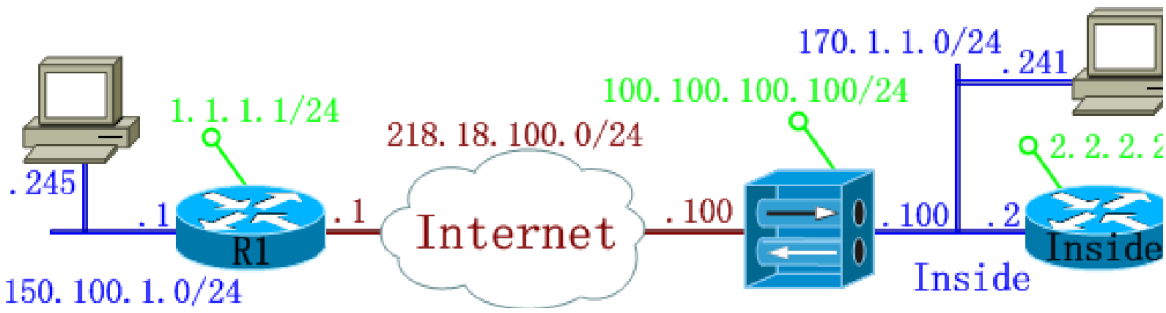
```
3) Save changes to Config file
4) Continue
5) Exit
Quick -> 2
This table shows current IP addresses.
  Intf         Status       IP Address/Subnet Mask          MAC Address
--------------------------------------------------------------------------
------
Ether1-Pri|      UP       |     10.1.1.100/255.255.255.0   |
00.90.A4.08.14.48
Ether2-Pub|Not Configured|        0.0.0.0/0.0.0.0          |
--------------------------------------------------------------------------
------
DNS Server(s): DNS Server Not Configured
DNS Domain Name:
Default Gateway: Default Gateway Not Configured
> Enter IP Address
Quick Ethernet 2 -> [ 0.0.0.0 ] 218.18.100.100   <===必须要在快速模式下配
置
                                                  重启 : ctrl+P  ===> PRB
> Enter Subnet Mask
Quick Ethernet 2 -> [ 255.255.255.0 ] 255.255.255.0
1) Ethernet Speed 10 Mbps
2) Ethernet Speed 100 Mbps
3) Ethernet Speed 10/100 Mbps Auto Detect
Quick Ethernet 2 -> [ 3 ]
1) Enter Duplex - Half/Full/Auto
2) Enter Duplex - Full Duplex
3) Enter Duplex - Half Duplex
Quick Ethernet 2 -> [ 1 ]
> MTU (68 - 1500)
Quick Ethernet 2 -> [ 1500 ]
1) Modify Ethernet 1 IP Address (Private)
2) Modify Ethernet 2 IP Address (Public)
3) Save changes to Config file
4) Continue
5) Exit
Quick -> 3
1) Modify Ethernet 1 IP Address (Private)
2) Modify Ethernet 2 IP Address (Public)
3) Save changes to Config file
4) Continue
5) Exit
Quick -> h       <===返回主菜单模式
```

1) Configuration
2) Administration
3) Monitoring
4) Save changes to Config file
5) Help Information
6) Exit
Main ->

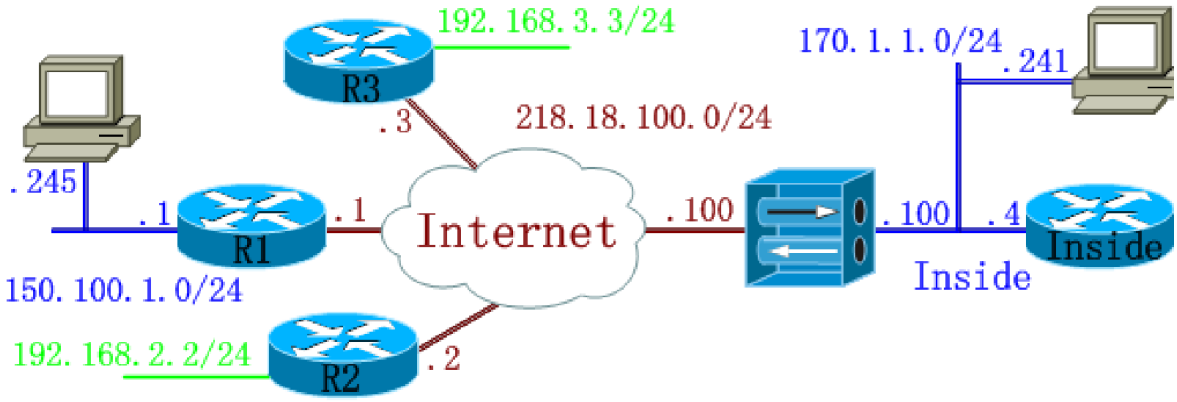
配置:
更改IP地址: 1 --> 1
更改主机名字: 1 --> 2 --> 6 --> 1 --> 1
配置路由(包含:默认路由): 1 --> 2 --> 3
删除默认路由: VPN3005: Routing -> [218.18.100.1] 0.0.0.0
查看路由: 3 --> 1
PING: 2 --> 5

实验 1:
The object of the experiment:

1. set private and public interface in GUI
2. create rules P200 <<VPN 完全配置指南>>
3. transfer rules in filter P202 <<VPN 完全配置指南>>
4. set route P279 <<VPN 完全配置指南>>
   The Router ID must be set before OSPF can be enabled.
5. set administrator and user P291 <<VPN 完全配置指南>>
6. set AAA server P292 <<VPN 完全配置指南>>



实验 2 :



软件客户端:

concentrator : 1. 建立特定组 P160 <<VPN 完全配置指南>>
组的详细配置 P83 <<安全虚拟专用网络>>
2. 建立地址池 P172 <<VPN 完全配置指南>>
3. 地址分配 P169 <<VPN 完全配置指南>>
4. 内部验证用户的建立(xauth user) P173 <<VPN 完全配置指南>>
5. 拨号后,"150.100.1.245" PING 向Inside路由器
注意: PING不通时,查看一下路由问题.
6. client 反向路由注入 P206 <<安全虚拟专用网络>>
7. set Local LAN Routes/Split Tunnel
8. view session/protocol/encrytion/ipsec

IPSec over TCP/NAT-T/IPSec over UDP: P218 <<安全虚拟专用网络>>

NAT-T/IPSec over UDP:

1. 在"NAT-T"和"IPSec over UDP"同时使用时,"NAT-T"优先
2. 协商阶段使用的还是UDP 500
3. 最后数据流把ESP封装进UDP的4500

IPSec over TCP:

1. 它是没有UDP 500协商包的
2. 它首先进行TCP的三次握手,建立好TCP连接后.所有的协商包都是在这个TCP中传输的

注意:只有软件client才支持"IPSec over TCP"和"IPSec over UDP"

硬件CLIENT端: <===和Easy VPN一样

注意:可能由于路由问题,R2不能"telnet"到"inside"路由器

测试:"R2"使用环回口"telnet"到"R3","R2"使用环回口"telnet"到"inside"路由器

网络扩展模式:

concentrator : 1. choose network extension mode in group P431 <<VPN完全配置指南>>

2. choose network extension reverse route injection
P434 <<VPN 完全配置指南>>

R2#show crypto ipsec client ezvpn <=====网络扩展模式没有分配地址是正常的
Easy VPN Remote Phase: 2
Tunnel name : cisco
Inside interface list: Loopback0,
Outside interface: Ethernet0/0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Split Tunnel List: 1
Address : 170.1.1.0
Mask : 255.255.255.0
Protocol : 0x0
Source Port: 0
Dest Port : 0

LAN-TO-LAN:
concentrator : 1. Policy/proposal P229 <<安全虚拟专用网络>>
2. transfer set P253 <<VPN 完全配置指南>>
3. 起用PFS P232 <<安全虚拟专用网络>> P265 <<VPN 完全配置指南>>
4. 注意:路由问题

WEB VPN: ------- 充当代理服务器的作用
OPEN SSL SERVICE:
condition : open http service in inner server
concentrator : 1. open SSL service in interface P216 <<VPN 完全配置指南>>

PORT FORWARDING:
concentrator : 1. port forwarding P225 <<VPN 完全配置指南>>
2. create Web server/CIFS server URL P222 <<VPN 完全配置指南>>
3. WebVPN label P228 <<VPN 完全配置指南>>

client : 1. Dial-up to concentrator
2. click "start application access"
3. "telnet 127.0.0.1 2323" in CLI
4. "show users" will be display concentrator's private interface

PKI

PKI ----- Public Key Infrastructure

证书生成　P108 <<安全虚拟专用网络>>

CA服务器可以是Windows Server,也可以是Router的IOS(12.4version以上)

C:\>mstsc /v: 170.100.1.241　　<===远程登录170.100.1.241server(in Windows command line)

CA 服务器(windows server)　　P222 <<网络安全基础>>

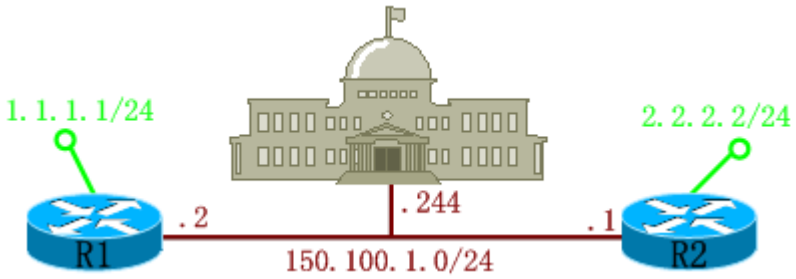
first of all : install Internet Information Services

and : Control Panel ====> Add/Remove Programs =====> Add/Remove Windows Components
=====> Certificate Services =====> Remote administration mode ===>
furthermore : Certification Authority =====> Security_CA(right-click) =====> Policy Module
======> Configure =====> Default Action
what is more :
1. set the certificate request status to pending administrator must explicitly issue the certificate
2. Always issue the certificate

标准的CA 服务器地址: http://170.100.1.118/certsrv

可支持CISCO设备的CA 服务器地址:http://170.100.1.118/certsrv/mscep/mscep.dll

实验 1:



set clock and NTP:

```
R1(config)#clock timezone GMT +8
R1#clock set 16:37:00 18 jun 2007
R1(config)#ntp master
```

R2(config)#clock timezone GMT +8
R2(config)#ntp server 150.100.1.1 <====NTP只同步时间,不同步时区

generate key:
R1(config)#ip domain-name cisco.com <===在高版本路由器上不用此命令,也可产生KEY
R1(config)#crypto key generate rsa usage-keys label IKE
Jun 18 11:18:43.615: %SSH-5-ENABLED: SSH 1.5 has been enabled

adjudgement CA:
R1(config)#crypto ca trustpoint R1 <===12.4版本以后,将"CA"改成了"PKI"
R1(ca-trustpoint)#enrollment url
http://150.100.1.244/certsrv/mscep/mscep.dll
R1(ca-trustpoint)#subject-name cn=R1
R1(ca-trustpoint)#crl optional
R1(ca-trustpoint)#rsakeypair IKE

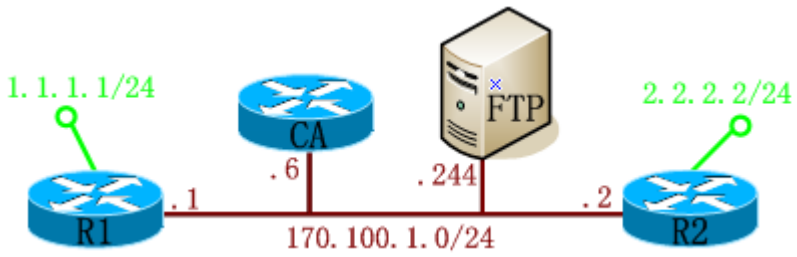
verify CA:
R1(config)#crypto ca authenticate R1

R1#show crypto ca certificates
R1#show crypto key mypubkey rsa

request my certificate:
R1(config)#crypto ca enroll R1

R2的配置与R1配置类似,然后再配置IKE的第一阶段和第二阶段

实验 2:



指定NTP SERVER:
IOS_CA(config)#clock timezone GMT +8
IOS_CA(config)#ntp server 170.100.1.1
R1和R2的配置和上一个实验一样

create certificate server:
IOS_CA(config)#ip http server
IOS_CA(config)#ip domain-name cisco.com

```
IOS_CA(config)#crypto key generate rsa usage-keys label IKE_CS
IOS_CA(config)#crypto pki server IKE_CS
IOS_CA(cs-server)#database url ftp://170.100.1.244
IOS_CA(cs-server)#database username ftp password cisco123
IOS_CA(cs-server)#database archive pem
IOS_CA(cs-server)#database level complete
IOS_CA(cs-server)#issuer-name cn=IOS_CA ou=IPSEC
IOS_CA(cs-server)#grant ?
  auto     Automatically grant incoming SCEP enrollment requests
  none     Automatically reject any incoming SCEP enrollment request
  ra-auto  Automatically grant RA-authorized incoming SCEP enrollment
request
IOS_CA(cs-server)#lifetime ?
  ca-certificate      Lifetime of the Certificate Server signing
certificate
  certificate         Lifetime of certificates issued by this Certificate
                      Server
  crl                 Lifetime of CRL's published by this Certificate
Server
  enrollment-request  Lifetime of an Enrollment Request
IOS_CA(cs-server)#no shutdown

IOS_CA#show crypto pki server
```

generate key:

```
R1(config)#ip domain-name cisco.com
R1(config)#crypto key generate rsa usage-keys label IKE
Jun 18 11:18:43.615: %SSH-5-ENABLED: SSH 1.5 has been enabled
```

adjudgement CA:

```
R1(config)#crypto ca trustpoint R1
R1(ca-trustpoint)#enrollment url http://170.100.1.6
Router(ca-trustpoint)#usage ike
Router(ca-trustpoint)#usage ?
  ike         IKE certificate
  ssl-client  SSL client certificate
  ssl-server  SSL server certificate
R1(ca-trustpoint)#subject-name cn=R1
R1(ca-trustpoint)#crl optional
R1(ca-trustpoint)#rsakeypair IKE
```

verify CA:

```
R1(config)#crypto ca authenticate R1
```

request my certificate:

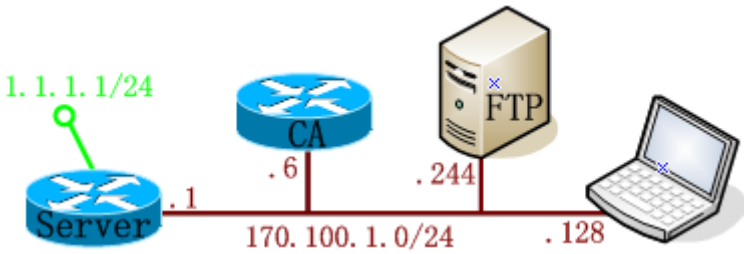
```
R1(config)#crypto ca enroll R1
R1(config)#CRYPTO_PKI:    Signing Certificate Reqeust Fingerprint:
7BEF0B2D BF995CAD 7175DE70 889A42BF
R1(config)#CRYPTO_PKI:    Encryption Certificate Request Fingerprint:
47C92D5C 8A82E568 2C716750 28692616
```

R2的配置与R1配置类似,然后再配置IKE的第一阶段和第二阶段

issue certificate:

```
IOS_CA#crypto pki server IKE_CS info requests

IOS_CA#crypto pki server IKE_CS grant all
```

实验 3:



服务器端的配置:

```
Server(config)#ip domain-name cisco.com

Server(config)#crypto key generate rsa usage-keys label IKE
Server(config)#crypto ca trustpoint R1
Server(ca-trustpoint)#enrollment url http://170.100.1.6
Server(ca-trustpoint)#usage ike
Server(ca-trustpoint)#subject-name cn=R1_VPN_SRV
Server(ca-trustpoint)#crl optional
Server(ca-trustpoint)#rsakeypair IKE
Server(config)#crypto ca authenticate R1
Server(config)#crypto ca enroll R1          <===不要输入密码
Server(config)#crypto isakmp policy 10
Server(config-isakmp)#hash md5
Server(config-isakmp)#authentication rsa-sig
Server(config-isakmp)#group 1      <===基于证书只能用"group 1",基于"Pre-
share key"只能用"group 2"
Server(config)#crypto isakmp identity dn
Server(config)#crypto isakmp identity ?
  address   Use the IP address of the interface for the identity
  dn        Use the distinguished name of the router cert for the
identity
  hostname  Use the hostname of the router for the identity
```

```
Server(config)#aaa new-model
Server(config)#aaa authentication login NOAU none
Server(config)#line con 0
Server(config-line)#login authentication NOAU
Server(config)#line vty 0 4
Server(config-line)#login authentication NOAU
Server(config)#aaa authentication login XAUTH local
Server(config)#username cisco password cisco123
Server(config)#aaa authorization network MODE local
Server(config)#crypto isakmp client configuration group EZVPN
Server(config-isakmp-group)#key cisco456        <===不起作用
Server(config-isakmp-group)#pool VPNPOOL
Server(config-isakmp-group)#acl 101
Server(config)#access-list 101 permit ip 1.1.1.0 0.0.0.255 any
Server(config)#crypto isakmp profile EZVPN
Server(conf-isa-prof)#match identity group EZVPN
Server(conf-isa-prof)#client authentication list XAUTH
Server(conf-isa-prof)#isakmp authorization list MODE
Server(conf-isa-prof)#client configuration address respond
Server(conf-isa-prof)#ca trust-point R1
Server(config)#crypto ipsec transform-set cisco esp-des esp-md5-hmac
Server(config)#crypto dynamic-map DMAP 10
Server(config-crypto-map)#set transform-set cisco
Server(config-crypto-map)#set isakmp-profile EZVPN
Server(config-crypto-map)#reverse-route
Server(config)#crypto map SMAP 10 ipsec-isakmp dynamic DMAP
Server(config)#interface e0/0
Server(config-if)#crypto map SMAP
Server(config)#ip local pool VPNPOOL 192.168.168.100 192.168.168.200
```

标准证书申请:

在客户端手动申请证书: P130 <<安全虚拟专用网络>>

标准IOS的证书申请:

与申请标准证书差不多

```
IOS_CA#crypto pki server IKE_CS request pkcs10 terminal

IOS_CA#crypto pki server IKE_CS info requests
Enrollment Request Database:
Subordinate CA certificate requests:
ReqID  State      Fingerprint                      SubjectName
--------------------------------------------------------------------

RA certificate requests:
ReqID  State      Fingerprint                      SubjectName
```

```
------------------------------------------------------------------
Router certificates requests:
ReqID  State      Fingerprint                         SubjectName
------------------------------------------------------------------
3      pending    2E6D1CA06E6366A37CE723CF8313849F
cn=IOS_CA,ou=EZVPN,o=CHINA,st=GZ
2      pending    636665F2FB3F1F4E26999CF64E8F619B serialNumber=18726165
+hostname=Server.cisco.com,cn=R1_VPN_SRV
1      pending    1F0B8C1FB9FE1B6C2DB4608B3FD2FAAD serialNumber=18726165
+hostname=Server.cisco.com,cn=R1_VPN_SRV
```

使用证书连接远程接入VPN:　　P137 <<安全虚拟专用网络>>

VPN Experiment

将switch的F0/11到F0/16的接口还原到默认配置:

```
Switch(config)#default interface range fastEthernet 0/11 - 16
```

VPN实验:

1. Lan to Lan :
   老命令
   新命令 :isakmp profile

2. IPSEC over GRE

3. GRE over IPSEC:
   老命令 : map
   新命令 : IPSEC profile

4. NAT-T , 应用层网关

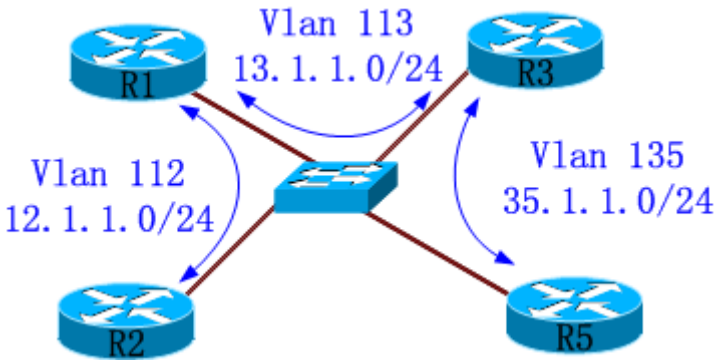
5. DMVPN

6. Redmdency VPN

7. EZVPN(RAVPN) <==|
   Sofe ware client <==|
   |=>Client mode <==|==课程录像(VPN实验指导第二天)
   HW==>| |
   |=>Network Extention <==|

192.168.16.253

Experiment 1:

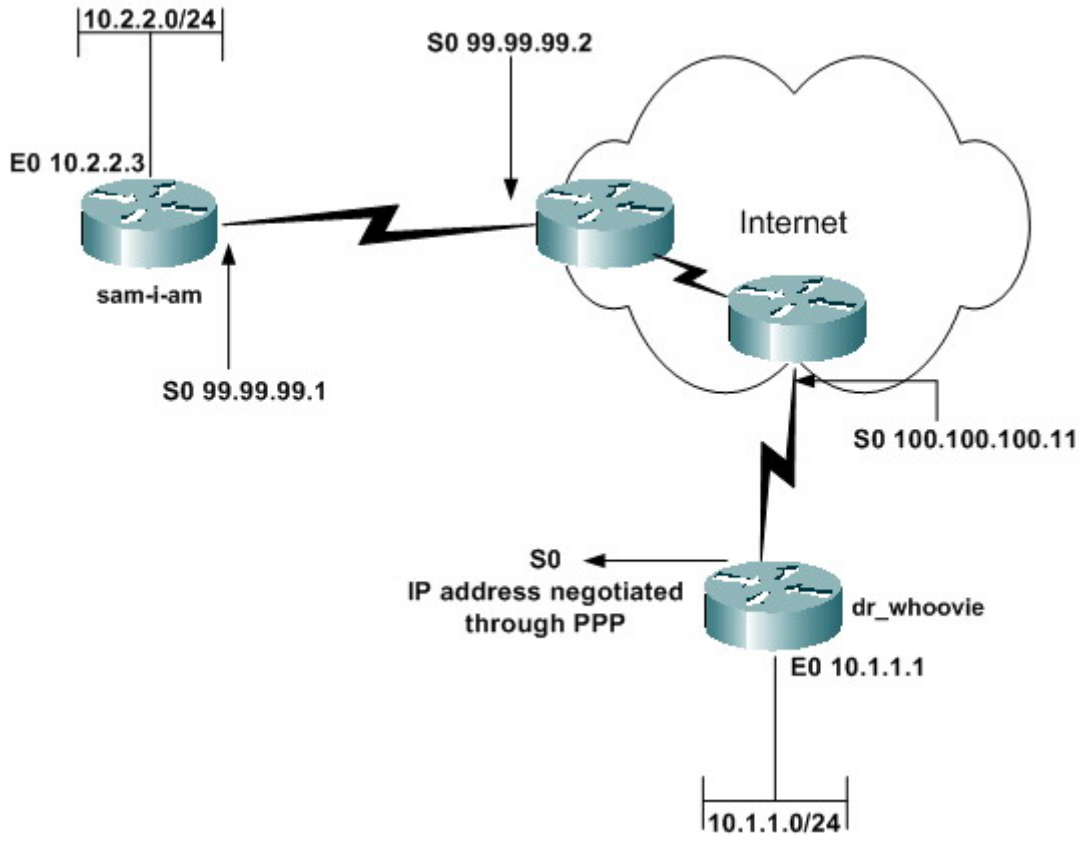



```
Switch(config)#vlan 112
Switch(config)#vlan 113
Switch(config)#vlan 135
Switch(config)#int f0/12
Switch(config-if)#switchport access vlan 112
Switch(config-if)#int f0/15
Switch(config-if)#switchport access vlan 135
Switch(config)#int f0/11
Switch(config-if)#switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk
Switch(config-if)#int f0/13
Switch(config-if)#switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk

R1(config)#int e0/0
R1(config-if)#no sh
R1(config)#interface Ethernet0/0.12
R1(config-subif)#encapsulation dot1Q 112
R1(config-subif)#ip address 12.1.1.1 255.255.255.0
R1(config)#interface Ethernet0/0.13
R1(config-subif)#encapsulation dot1Q 113
R1(config-subif)#ip address 13.1.1.1 255.255.255.0
```

```
R2(config-line)#int e0/0
R2(config-if)#ip add 12.1.1.2 255.255.255.0
R2(config-if)#no sh

R3(config)#int e0/0
R3(config-if)#no sh
R3(config)#interface Ethernet0/0.13
R3(config-subif)#encapsulation dot1Q 113
R3(config-subif)#ip address 13.1.1.3 255.255.255.0
R3(config)#interface Ethernet0/0.35
R3(config-subif)#encapsulation dot1Q 135
R3(config-subif)#ip address 35.1.1.1 255.255.255.0

R5(config)#int e0
R5(config-if)#ip add 35.1.1.5 255.255.255.0
R5(config-if)#no sh
```

Experiment 2:

VPN实例配置方案一中文详细注解(一)<图>



Router:sam-i-am(VPN Server)

```
Current configuration:
!
version 12.2
service timestamps debug uptime
service timestamps log up time
no service password-encryption
!
hostname sam-i-am
!
ip subnet-zero

!--- IKE配置

sam-i-am(config)#crypto isakmp policy 1 //定义策略为1
sam-i-am(isakmp)#hash md5 //定义MD5散列算法
sam-i-am(isakmp)#authentication pre-share //定义为预共享密钥认证方式
sam-i-am(config)#crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0

!--- 配置预共享密钥为cisco123,对等端为所有IP

!--- IPSec协议配置

sam-i-am(config)#crypto ipsec transform-set rtpset esp-des esp-md5-
hmac

!--- 创建变换集 esp-des esp-md5-hmac

sam-i-am(config)#crypto dynamic-map rtpmap 10 //创建动态保密图rtpmap
10
san-i-am(crypto-map)#set transform-set rtpset //使用上面的定义的变换
集rtpset
san-i-am(crypto-map)#match address 115 //援引访问列表确定受保护的流量
sam-i-am(config)#crypto map rtptrans 10 ipsec-isakmp dynamic rtpmap

!--- 将动态保密图集加入到正规的图集中

!
interface Ethernet0
ip address 10.2.2.3 255.255.255.0
no ip directed-broadcast
ip nat inside
```

```
no mop enabled
!
interface Serial0
ip address 99.99.99.1 255.255.255.0
no ip directed-broadcast
ip nat outside
crypto map rtptrans //将保密映射应用到S0接口上

!
ip nat inside source route-map nonat interface Serial0 overload
!--- 这个NAT配置启用了路由策略，内容为10.2.2.0到10.1.1.0的访问不进行
地址翻译
!--- 到其他网络的访问都翻译成S0接口的IP地址

ip classless
ip route 0.0.0.0 0.0.0.0 Serial0 //配置静态路由协议
no ip http server
!
access-list 115 permit ip 10.2.2.0 0.0.0.255 10.1.1.0 0.0.0.255
access-list 115 deny ip 10.2.2.0 0.0.0.255 any
!
access-list 120 deny ip 10.2.2.0 0.0.0.255 10.1.1.0 0.0.0.255
access-list 120 permit ip 10.2.2.0 0.0.0.255 any
!
sam-i-am(config)#route-map nonat permit 10 //使用路由策略
sam-i-am(router-map)#match ip address 120
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
!
end

Router:dr_whoovie(VPN Client)

Current configuration:
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
```

```
!
hostname dr_whoovie
!
ip subnet-zero
!
dr_whoovie(config)#crypto isakmp policy 1 //定义策略为1
dr_whoovie(isakmp)#hash md5 //定义MD5散列算法
dr_whoovie(isakmp)#authentication pre-share //定义为预共享密钥认证方
式
dr_whoovie(config)#crypto isakmp key cisco123 address 99.99.99.1

!--- 配置预共享密钥为cisco123,对等端为服务器端IP99.99.99.1

!--- IPSec协议配置

dr_whoovie(config)#crypto ipsec transform-set rtpset esp-des esp-md5-
hmac

!--- 创建变换集 esp-des esp-md5-hmac

dr_whoovie(config)#crypto map rtp 1 ipsec-isakmp

!--- 使用IKE创建保密图rtp 1

dr_whoovie(crypto-map)#set peer 99.99.99.1 //确定远程对等端
dr_whoovie(crypto-map)#set transform-set rtpset //使用上面的定义的变
换集rtpset
dr_whoovie(crypto-map)#match address 115 //援引访问列表确定受保护的流
量

!
interface Ethernet0
ip address 10.1.1.1 255.255.255.0
no ip directed-broadcast
ip nat inside

no mop enabled
!
interface Serial0
ip address negotiated //IP地址自动获取
no ip directed-broadcast
ip nat outside
encapsulation ppp //S0接口封装ppp协议
no ip mroute-cache
```

```
no ip route-cache
crypto map rtp //将保密映射应用到S0接口上

!
ip nat inside source route-map nonat interface Serial0 overload
!--- 这个NAT配置启用了路由策略，内容为10.1.1.0到10.2.2.0的访问不进行地址翻译
!--- 到其他网络的访问都翻译成S0接口的IP地址

ip classless
ip route 0.0.0.0 0.0.0.0 Serial0 //配置静态路由协议
no ip http server

!
access-list 115 permit ip 10.1.1.0 0.0.0.255 10.2.2.0 0.0.0.255
access-list 115 deny ip 10.1.1.0 0.0.0.255 any

access-list 120 deny ip 10.1.1.0 0.0.0.255 10.2.2.0 0.0.0.255
access-list 120 permit ip 10.1.1.0 0.0.0.255 any

!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
route-map nonat permit 10 //使用路由策略
match ip address 120
!
line con 0
transport input none
line aux 0
line vty 0 4
password ww
login
!
end
```

----------IKE配置----------

IPSec VPN对等端为了建立信任关系，必须交换某种形式的认证密钥。

Internet 密钥交换(Internet Key Exchange，IKE)是一种为IPSec管理和交换密钥的标准方法。

一旦两个对等端之间的IKE协商取得成功，那么IKE就创建到远程对等端的安全关联(security association，SA)。SA是单向的；在两个对等端之间存在两个SA。

IKE使用UDP端口500进行协商，确保端口500不被阻塞。

配置

1、（可选）启用或者禁用IKE

(global)crypto isakmp enable
或者
(global)no crypto isakmp enable
默认在所有接口上启动IKE

2、创建IKE策略略

(1)定义策略

(global)crypto isakmp policy priority
注释：policy 1表示策略1，假如想多配几个VPN，可以写成policy 2、policy3
---

(2)（可选）定义加密算法

(isakmp)encryption {des | 3des}
加密模式可以为56位的DES-CBC(des，默认值)或者168位的3DES(3des)

(3)（可选）定义散列算法

(isamkp)hash {sha | md5}
默认sha

(4)（可选）定义认证方式

(isamkp)authentication {rsa-sig | rsa-encr | pre-share}
rsa-sig 要求使用CA并且提供防止抵赖功能；默认值
rsa-encr 不需要CA，提供防止抵赖功能
pre-share 通过手工配置预共享密钥

(5)（可选）定义Diffie-Hellman标识符

(isakmp)group {1 | 2}
注释：除非购买高端路由器，或是VPN通信比较少，否则最好使用group 1长度的密钥，group命令有两个参数值：1和2。参数值1表示密钥使用768位密钥，

参数值2表示密钥使用1024位密钥，显然后一种密钥安全性高，但消耗更多的

CPU时间。

(6)（可选）定义安全关联的生命期

(isakmp)lifetime seconds
注释：对生成新SA的周期进行调整。这个值以秒为单位，默认值为86400，也就是一天。值得注意的是两端的路由器都要设置相同的SA周期，否则VPN在正

常初始化之后，将会在较短的一个SA周期到达中断。

手把手指南：教你如何架设VPN
简介

让远程用户连接Exchange Server的传统解决方案是使用Outlook Web Access. 然而，为何不使用虚拟专用网（Virtual Private Network，VPN）让你的远程用户与你的Exchange连接在一起呢？

如果你不熟悉VPN，我将向你介绍，VPN是穿越不安全的网络，譬如Internet的一个逻辑的、安全的网络连接。远程用户能够通过他们的已经存在的Internet连接，安全地连接进入你的网络，就像他们亲自在办公室中一样。另外一个优势是，VPN是交换独立的，这意味着你能够使用VPN连接访问Exchange Server，而不用考虑版本问题，并且你还可以使用VPN连接访问其它网络资源。

VPN技术对机构和很多远程用户来说，是极端有用的，但在设定上，它可能有些复杂。下面的指南将手把手的教你如何建立VPN，它包含各个步骤详细的操作流程。

第一步：系统需求

VPN分为两种，一种是硬件解决方案，一种是软件解决方案，在这个手把手的指南中，我将介绍一种软件解决方案，即使用Microsoft产品建立VPN.

为了架设VPN，你将需要三个独立的Windows 2003服务器和至少一个远程用户，远程用户的机器上需要运行Windows XP操作系统。

你的VPN需要的第一台Windows 2003服务器是一台基本的基础设施服务器，它必须作为一台域控制器（domain controller），DHCP服务器（DHCP server），DNS服务器（DNS Server）和认证中心（certificate authority）。如果你的网络中已经有一台Windows 2003服务器，你就不需要去购买一台服务器担当此角色。

任何Windows 2003域都至少有一台域控制器和一台作为DNS的服务器，多数Windows 2003网络同时运行DHCP服务。如果你所有的这些服务已经到位，你所关心的唯一的事情就是设置一个认证中心（我将在第三步为你说明如何做这件事情）。下载，你只需知道作为认证中心的那台服务器必需运行Windows Server 2003 Enterprise Edition操作系统。

你需要的第二台服务器将是VPN服务器（VPN server），Windows Server 2003 Standard Edition和Enterprise Edition都提供了VPN服务器的必要软件，因此，你不需要在这台服务器上安装任何特别的软件。唯一特别的是硬件上，这台服务器需要双网卡，一块网卡连接Internet，另一块网卡则连接你的专用企业网络。

你需要的最后一台服务器将是认证服务器（authentication server）。当远程用户通过VPN尝试进入你的企业网络时，他们必需通过认证。远程用户认证的机制可以选择RADIUS服务器（RADIUS server），RADIUS 是Remote Authentication Dial In User Service（远程身份验证拨入用户服务）的首字母缩写。在Windows Server 2003 Standard Edition和Enterprise Edition中，包含有微软自有版本的RADIUS. 微软的RADIUS叫做Internet验证服务（Internet Authentication Service，IAS），对这台服务器来说，没有特殊的硬件和软件要求。

在这一部分，最后我想说的是服务器的安置问题。任何一台我谈论到的服务器都将通过Hub或交换机接入你的专用网络，唯一与外界连接的服务器是你的VPN服务器，但将VPN服务器直接与Internet连接将会带来安全风险，因此，在VPN服务器的前面放置一台防火墙是很好的解决办法，你可以用它过滤掉除了VPN通讯外所有其它的信息。

在第二步，我们将开始配置域的过程，所以在进入下一步之前，你的网络中必需包含必需的Windows 2003域控制器和DNS服务器。

第二步：实施DHCP服务

1.打开服务器的控制面板，选择“添加或删除程序”。

2.当“添加或删除程序”对话框出现时，点击“添加/删除Windows组件”按钮。

3.在弹出的窗口中，选择“网络服务”，按下“详细信息”。

4.现在从网络服务列表中选择“动态主机配置协议（DHCP）”，然后单击“确定”，进行下一步操作。

Windows现在将安装DHCP服务，安装结束后，你将要创建一个地址范围，并且启动DHCP服务器，在你的网络上运行。

5.为了做到这些，请在控制面板－－管理工具中选择动态主机配置协议（DHCP）配置，打开DHCP管理器。

6.在DHCP管理器中你的服务器上单击右键，选择启动（Authorize）。

7.启动DHCP服务器后，在DHCP管理器的服务器列表窗口中单击右键，选择“新

建作用域（New Scope）”，这将启动新建作用域向导。

8.点击下一步略过向导的欢迎界面。

9.输入你正在创建的作用域的名称，并且点击下一步。（你可以输入任何你想到的名称，但在这个教程中，我将命名此作用域为“Corporate Network”。）

10.现在你将需要填入IP地址范围。在这里只需输入你已经使用的起始IP地址和结束IP地址，但注意不要与已经存在的IP地址冲突。长度和子网掩码部分则会自动输入，不需要你的干涉，当然，你也可以手动调节这两者的值。

11.接下来的三个画面包括一些你不必关心的设置，连续三次点击下一步，直到你进入“路由（默认网关）（Router （Default Gateway））”界面。

12.输入你网络网关的IP地址，点击添加，然后下一步。

13.输入你的域的名称和你的DHCP服务器的IP地址（IP address of your DHCP server），然后点击下一步。

14.单击下一步略过WINS配置窗口。

15.最后，根据提示选“是，我想激活作用域（Yes， I Want To Activate The Scope Now）”再点击“完成”即可结束最后设置。

第三步：创建一个企业认证中心

在我向你讲述如何创建一个企业认证中心之前，我将告诉你几个必需注意的事项。安装认证中心并不是一个轻松的过程，如果一个未经授权的用户进入了你的认证中心，他将几乎控制你的所有网络。同样，如果认证中心服务器当机，它可能对给你的网络带来毁灭性的破坏。

所以，一定要像保护原子弹一样保护你的认证中心，确保认证中心尽可能的安全，并频繁的做好全系统的备份，你还需要保护这些备份，以防止它们偶然地出现问题。下面是创建企业认证中心的具体过程。

1. 打开服务器的控制面板，选择“添加或删除程序”，点击其中的“添加/删除Windows组件”按钮。

2.选择Windows组件中的“证书服务”。

3.你将会看到一个警告窗口，上面的信息为：“安装证书服务后，计算机名和域成员身份都不能更改，因为计算机名到CA信息的绑定存储在Active Directory中。更改计算机名或域成员身份将使此CA颁发的证书无效。在安装证书服务前请确认配置了正确的计算机名和域成员身份。您想继续吗？”点击“是”，接受这一警

告信息，并点击“下一步”，开始安装证书服务。

4.选择“独立根CA”作为你想安装的CA类型，并点击下一步。

在这里，为自己的CA服务器取个名字，设置证书的有效期限。默认的证书有效期限为5年，不过你可以通过企业安全策略来增加或减少此有效期限。

5.填写好这两个文本框，点击下一步，Windows将开始生成加密密钥。

6.最后指定证书数据库和证书数据库日志的位置，按照默认即可，除非你自己想更换路径，然后点击下一步。

7.现在将出现一则消息，提示Windows必需重新启动IIS服务，才能够让证书服务正常运行。点击“是”，Windows将安装必要的组件。

第四步：安装Internet验证服务

Internet验证服务是Windows Server 2003实施RADIUS的一种服务，Internet验证服务将认证那些通过VPN连接进入你的企业网络的用户，因此，你的Internet验证服务器必需是你的域服务器中的一员，并且运行Windows Server 2003操作系统。为了正确安装IAS，请遵循以下的步骤：

1.定位到开始 | 设置 | 控制面板（Start| Settings | Control Panel）。

2.双击添加或删除程序（Add/Remove Programs）。

3.选择添加/删除Windows组件（Add/Remove Windows Components）。

4.在组件列表中，选择网络服务（Networking Services），并点击详细内容。

5.选择Internet验证服务（Internet Authentication Service）的确认框，然后点击OK，并点击下一步。

完成安装之后，系统中将具有用于因特网认证服务的管理工具的一个新的连接。接着，你必须为每一台机器设置一个客户端，并指定一个远距离访问规范以控制访问。

第五步：配置Internet验证服务

1.进入管理工具（Administrative Tools）-> Internet验证服务（Internet Authentication Service）。

2.在这里，你需要做的第一件事情是在活动目录（Active Directory）中注册你的Internet验证服务器。为了做到这些，请在Internet验证服务器（本地）

（Internet Authentication Service （Local））容器上单击右键，选择在活动目录中注册服务器（Register Server in Active Directory）。

3.点击确定完成注册过程。

4.现在，在RADIUS客户（RADIUS Clients）容器上单击右键，选择新RADIUS客户（RADIUS Clients）。如果你正好直到你某台客户端机器的IP地址或DNS名称，继续下去，输入一个友好的名字。否则，暂时将它留空，在随后的设置客户端连接时再进行填写。

5.点击下一步。

6.此时，会提示你输入一个共享的密钥。共享密钥是RADIUS服务器和客户端同时使用的密钥，确定客户端供应商选项设置为RADIUS标准，输入一个共享密钥值，点击完成。

第六步：创建远程访问策略

1. 在Internet验证服务控制台，右击远程访问策略（Remote Access Policies）容器，选择新建远程访问策略（New Remote Access Policy）选项，这将启动新建远程访问策略向导。

2. 在欢迎使用新建远程访问策略向导页，点击下一步。

3.在策略配置方式页，选择使用向导为通用环境建立典型的策略（Typical Policy for a Common Scenario）选项，在策略名字文本框中输入一个名字，在此我们命名为“VPN Access”，点击下一步。

4.在访问方式页，选择VPN 选项，然后点击Next.

5.在访问的组或者用户页，选择组或用户，然后点击添加。如果你还没有做这一步，我建议你花一些事件创建一个建立在能够通过VPN访问网络的用户纸上的活动目录组，然后将这个组添加进入策略。

6.点击下一步，进入认证方法窗口。

7.确认选择了“Microsoft Encrypted Authentication version 2 （MS CHAPV2） ”，然后点击下一步。

8.在策略加密级别页，确认只选择了“Strong encryption”选项，随后点击下一步，根据向导，在完成新建远程访问策略向导页点击完成。

第七步：配置VPN服务器

1.开始，请打开服务器的网络连接（Network Connections）目录，并将连接重命名为有意义的名字，例如，你可以将连接命名为企业和Internet，或者其它你喜欢的名字。

2.进入管理工具（Administrative Tools）->路由和远程访问（Routing and Remote Access），打开路由和远程访问管理器。

3.在管理器目录数中，右键单击你的VPN服务器，选择配置和启用路由和远程访问（Configure and Enable Routing and Remote Access），这将载入路由和远程访问服务器设置向导（Routing and Remote Access Server Setup Wizard）。

4.点击下一步，略过向导的欢迎页面，然后你将看到向导的配置窗口。

5.选择远程访问（拨号或VPN）Remote Access （Dial-Up or VPN），然后点击下一步。

6.选中VPN前面的复选框，点击下一步。

7.现在，你将看到一个窗口，此窗口中显示有你的机器的网络连接。选择连接Internet的那个连接，确认启用安全（Enable Security）复选框被选择，然后单击下一步。

8.确认选中了自动（Automatically），并点击下一步。

9.现在，在选项中选择和设置跟RADIUS服务器一起工作的服务器，并单击下一步。

10.输入你的RADIUS服务器的IP地址，和你为RADIUS服务器分配的共享密钥信息。

11.点击下一步，点击完成。

第八步：使VPN服务器和DHCP服务器关联起来

1.在路由和远程访问控制台目录数中指向你的服务器->IP路由（IP Routing）->DCHP中继代理（DHCP Relay Agent）。

2.在DHCP中继代理上单击右键，选择属性（Properties）。

3.输入你的DHCP服务器的IP地址，点击添加，然后再单击确定。

现在你的VPN服务器已经配置好了。万事俱备，剩下的唯一要做的就是配置你的客户端，使它们与你刚刚创建的VPN服务协同工作。

第九步：配置远程客户端

你应该可以记起，我们必需为那些能够通过VPN访问企业内部网络的用户创建一个特别的安全组。所以，我假设你的远程用户已经被加入必要的组，并且客户端的计算机已经接入了Internet.

允许一台运行Windows XP操作系统的客户端计算机访问你的专用网络，就必需告诉它们如何使用VPN连接。

1.为了做到这些，请打开控制面板（Control Panel），选择网络和Internet连接（Network and Internet Connections）选项。

2.创建一个新连接，在向导中选择连接到我的工作场所的网络（Connection to the Network At Your Workplace）选项。

3.Windows现在将询问你，想创建一个拨号连接（dial-up connection），还是一个虚拟专用网络连接（VPN connection）。选择虚拟专用网络连接，点击下一步。

4.在这一步，你将看到公司名的项目，你能够在这里输入你公司的名字，你连接的服务器的名字，或者其它你用来描述连接的东西。

5.点击下一步，你将被要求输入你正连接的计算机的主机名或IP地址，请填入你的VPN服务器的外网IP地址（即连接进入Internet的IP地址）。

6.再次单击下一步 ，根据向导最后完成你的VPN连接创建。

第十步：测试客户端连接

1.双击可用连接列表中的VPN连接。

2.你将被要求输入用户名和密码。为了更好的使用VPN连接，我们还需要进行一些设置，因此请点击此界面中的属性（Properties）按钮。

3.在属性窗口中，选择网络（Networking）标签。

4.选择VPN类型为PPTP VPN，按下确定按钮。

5.现在你将回到VPN连接登陆窗口，以domain/username格式输入你的用户名 .

6.然后输入你的密码，点击连接（Connect）。

7.这里可能会提供一个机会，让你选择想连接那个网络。如果有提示，选择局域网连接（LAN Connection）选项。

8.一旦连接建立，请在开始->运行中输入\\servername\ROOT命令。

你应该可以看到你的服务器的C盘的内容（假设你有相应的权限）。当然，这种直接访问服务器C盘的方式非常罕见，多数情况下，你只能访问服务器上的特定共享资源，而要做到这些，你应该在开始->运行中输入\\servername\sharename.

第十一步：改变VPN的配置选项

PIX

Initial/ospf/rip/static

注意: PIX上没有CDP功能

```
pixfirewall(config)#clear config all          <===清空所有配置(不需要重启)
pixfirewall(config)#clear config inter e0/0
pixfirewall(config)#clear config nat
pixfirewall#wr                  <====存盘
pixfirewall(config)#wr erase      <====清空start-config

pixfirewall# show version
Cisco PIX Security Appliance Software Version 7.2(2)       <====此PIX为
7.22版
Device Manager Version 5.2(2)
Compiled on Wed 22-Nov-06 14:16 by builders
System image file is "flash:/pix722.bin"
Config file at boot was "startup-config"
pixfirewall up 5 hours 27 mins
Hardware:   PIX-515, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB                          <====Flash的大小为
16M
BIOS Flash AT29C257 @ 0xfffd8000, 32KB
 0: Ext: Ethernet0           : address is 0050.54ff.25e9, irq 10
 1: Ext: Ethernet1           : address is 0050.54ff.25ea, irq 7
 2: Ext: Ethernet2           : address is 00d0.b791.1eed, irq 11
Licensed features for this platform:
Maximum Physical Interfaces : 6
Maximum VLANs               : 25
Inside Hosts                : Unlimited
Failover                    : Active/Active
VPN-DES                     : Enabled        <====此PIX支持VPN的DES
VPN-3DES-AES                : Disabled          <====此PIX不支持VPN的3DES
Cut-through Proxy           : Enabled
```

```
Guards                          : Enabled
URL Filtering                   : Enabled
Security Contexts               : 2        <===支持2个虚拟firewall+1个管理
firewall
GTP/GPRS                        : Disabled
VPN Peers                       : Unlimited

This platform has an Unrestricted (UR) license.

Serial Number: 480020141
Running Activation Key: 0xf2698e75 0x6954763e 0x9c011711 0xfcbe7668  <===
此KEY用于升级时使用(不能丢)
Configuration last modified by enable_15 at 18:13:54.995 UTC Sat Feb 3
2007
pixfirewall#
```

防火墙的种类　　P13 <<安全PIX防火墙>>

无限制/受限/故障倒换　　P36 <<安全PIX防火墙>>

命令NTP　P84 <<安全PIX防火墙>>

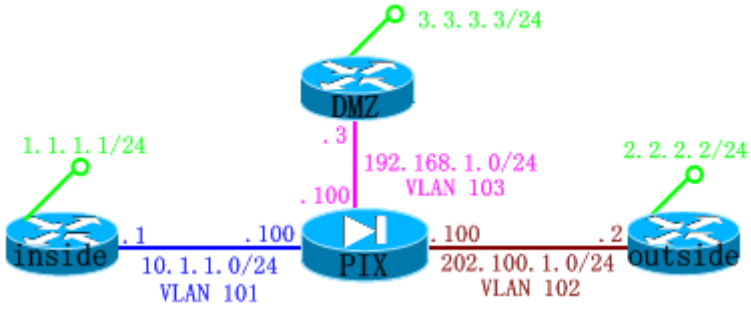
命令Syslog　　P93 <<安全PIX防火墙>>

Transparent Firewall

```
PIX(config)# access-list BPDUALLOW ethertype permit bpdu
PIX(config)# access-group BPDUALLOW in interface inside
PIX(config)# access-group BPDUALLOW in interface outside
Switch#debug spanning-tree bpdu
13:49:44: STP: VLAN0001 Fa0/22 tx BPDU: config protocol=ieee
    Data : 0000 00 00 00 80010008E3381700 00000000 80010008E3381700 8016
0000 1400 0200 0F00
Switch#show spanning-tree interface f0/22 detail
 Port 22 (FastEthernet0/22) of VLAN0001 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.22.
   Designated root has priority 32769, address 0008.e338.1700
   Designated bridge has priority 32769, address 0008.e338.1700
   Designated port id is 128.22, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   BPDU: sent 987, received 0
```

Multi context mode

实验 1 ： 配置OSPF/STATIC/RIP



配置子接口:

```
PIX(config)# interface e0.192
PIX(config-subif)# vlan 192
PIX(config-subif)# nameif outside
PIX(config-subif)# ip add 192.168.1.100 255.255.255.0
```

Static route: P239 <<安全PIX防火墙>>

```
PIX(config)# route outside 2.2.2.2 255.255.255.255 202.100.1.2
PIX(config)# route inside 1.1.1.1 255.255.255.255 10.1.1.1
PIX(config)# route DMZ 3.3.3.3 255.255.255.255 192.168.1.3
```

PIX# clear route <====> Router#clear ip route *

```
PIX(config)# show route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
S    1.1.1.1 255.255.255.255 [1/0] via 10.1.1.1, inside
S    2.2.2.2 255.255.255.255 [1/0] via 202.100.1.2, outside
S    3.3.3.3 255.255.255.255 [1/0] via 192.168.1.3, DMZ
C    202.100.1.0 255.255.255.0 is directly connected, outside
C    10.1.1.0 255.255.255.0 is directly connected, inside
C    192.168.1.0 255.255.255.0 is directly connected, DMZ
```

```
PIX(config)# clear configure route outside
PIX(config)# show run route
route inside 1.1.1.1 255.255.255.255 10.1.1.1 1
route DMZ 3.3.3.3 255.255.255.255 192.168.1.3 1
```

```
PIX(config)#clear config route                    <====清除route的相关配置
-------------------------------------------------
```

路由的负载均衡:　　基于同一个源目地址,不能实现负载均衡,因为它基于流的;基于不同目的地址，也不能实现负载均衡

```
PIX(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
PIX(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.2
PIX(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.3
```

Default route :　　最多只能定义3个默认路由(不包括tunnel的默认路由)

```
PIX(config)# route outside 0 0 202.100.1.1
PIX(config)# route outside 0 0 202.100.1.2
PIX(config)# route outside 0 0 202.100.1.3
PIX(config)# route inside 0 0 10.1.1.1 tunneled     <===这条默认路由只对解密后的流量起作用,不对明文流量起作用
PIX(config)# show route
C    202.100.1.0 255.255.255.0 is directly connected, outside
C    10.1.1.0 255.255.255.0 is directly connected, inside
C    192.168.1.0 255.255.255.0 is directly connected, DMZ
S*   0.0.0.0 0.0.0.0 [1/0] via 202.100.1.1, outside
                   [1/0] via 202.100.1.2, outside
                   [1/0] via 202.100.1.3, outside
S    0.0.0.0 0.0.0.0 [255/0] via 10.1.1.1, inside tunneled
```

配置OSPF

```
PIX(config)# router ospf 1
PIX(config-router)# network 192.168.1.0 255.255.255.0 area 0   <===PIX上是正掩码,路由器上是反掩码
-----------------------------------------------------------------
PIX(config)# show route
O    3.3.3.3 255.255.255.255 [110/11] via 192.168.1.3, 0:04:20, DMZ
C    202.100.1.0 255.255.255.0 is directly connected, outside
C    10.1.1.0 255.255.255.0 is directly connected, inside
C    192.168.1.0 255.255.255.0 is directly connected, DMZ
-----------------------------------------------------------------
PIX(config)# show ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
Interface
3.3.3.3           1   FULL/BDR        0:00:37     192.168.1.3     DMZ
-----------------------------------------------------------------
PIX(config)# show ospf database
       OSPF Router with ID (202.100.1.100) (Process ID 1)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link
```

```
count
3.3.3.3         3.3.3.3         593         0x80000003 0xb303 2
202.100.1.100   202.100.1.100   604         0x80000002 0x2939 1
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.100   202.100.1.100   603         0x80000001 0x8baa
-----------------------------------------------------------------
PIX(config)# show ospf interface
DMZ is up, line protocol is up
  Internet Address 192.168.1.100 mask 255.255.255.0, Area 0
  Process ID 1, Router ID 202.100.1.100, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 192.168.1.3
  Backup Designated router (ID) 202.100.1.100, Interface address
192.168.1.100
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 0:00:05
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 3.3.3.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
-----------------------------------------------------------
PIX(config-subif)# show route dmz
O    3.3.3.3 255.255.255.255 [110/11] via 192.168.1.3, 0:00:29, DMZ
C    192.168.1.0 255.255.255.0 is directly connected, DMZ
```

清除OSPF进程:

```
PIX(config)# clear ospf process
```

OSPF的验证:

区域验证:

```
PIX(config)# router ospf 1
PIX(config-router)# area 0 authentication
PIX(config-router)# interface e0.192
PIX(config-subif)# ospf message-digest-key 1 md5 cisco
```

链路验证:

```
PIX(config-subif)# int e0.192
PIX(config-subif)# ospf authentication message-digest
PIX(config-subif)# ospf message-digest-key 1 md5 cisco
```

OSPF的重分布:

```
PIX(config)# access-list HUAWEI permit host 1.1.1.1
PIX(config)# route-map wolf
PIX(config-route-map)# match ip address HUAWEI
PIX(config-router)# redistribute static route-map wolf subnets
PIX(config-router)# redistribute connected subnets
```

清除OSPF的所有配置:

```
PIX(config)# clear configure router ospf
PIX(config)# show run router ospf        <===查看OSPF的配置
```

管理接口:

```
pixfirewall(config-if)# management-only      <===不允许穿越流量,只允许抵达流量
pixfirewall(config-if)# no management-only
```

放行PING流量:

第1种: 写访问列表

第2种:

```
pixfirewall(config-if)# policy-map global_policy
pixfirewall(config-pmap)# class inspection_default
pixfirewall(config-pmap-c)#inspect icmp
```

个人总结的asa/pix 7.x透明模式防火墙的关键知识点

针对asa和pix的7.x
透明模式只支持两个接口，透明模式也可以用multi-context
注意透明模式没有nat,没有路由

1. 3层流量要明确放行（ospf，eigrp），要想跨防火墙建邻居，两边都要permit
2. 直连的outside和inside网络必须属于相同子网
3. 必须要配置一个网管ip地址,必须~~~
4 网管ip和内外网ip在同一段.
5. 管理ip不能做内网网关
6. 可以配置网关,但是只做网管用，远程访问防火墙用
7. 每个接口必须在不同vlan,（这个是看的资料上写的，暂时不是很理解）
8. 所有流量都可由ip acl和ethernet acl控制是否放行，eth acl只能管二层流量，但是如果deny any了，则 2，3层都不过
9. arp不需要放行就可以过去,除arp外，所有二层流量默认都不通
10. cdp不可以过

透明模式不支持,nat, dynamic routing protocol,ipv6, dhcp relay,qos, multicast, 不能终结vpn

NAT/PAT

〈端口地址转换〉

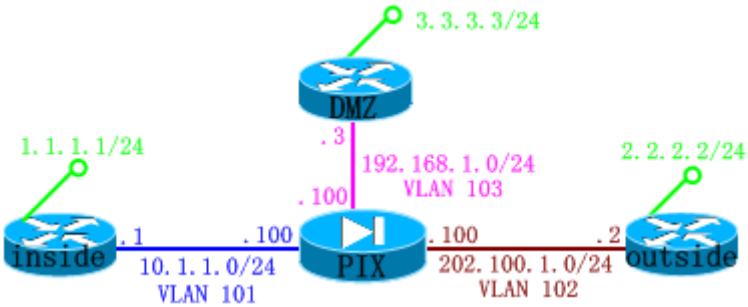
1. NAT

NAT对内部主机的IP地址分配是一对一的.

2. PAT

PAT允许内部多个主机对应一个IP地址,这个过程被称为超载(over loading),允许转化TCP连接或UDP会话的每个源端口号,这意味着内部每个主机被分配一个惟一的端口号,而这些主机使用同一个IP地址.

实验 1 :



PIX(config)# nat-control <===使用此命令后,高安全级别要"telnet"到低安全级别,必须要有转换项."nat-control"命令不对相同安全级别的接口互访起作用.

在允许相同级别相互时,默认不允许访问别的安全级别(不论高低):

PIX(config)# same-security-traffic permit inter-interface　　<====不同接口,相同安全级别的设备通信
PIX(config)# same-security-traffic permit intra-interface　　<====相同接口,不同的设备之间通信

使用以下命令,允许在访问相同级别的同时,又允许访问安全级别低的接口:
PIX(config)# static (inside,outside) 202.100.1.100 10.1.1.1

使用访问列表,允许低安全级别访问高安全级别,必需要有转换项:　　P236<<安全PIX防火墙>>
PIX(config)# no nat-control
PIX(config)# access-list OU permit ip any any
PIX(config)# access-group OU in interface outside

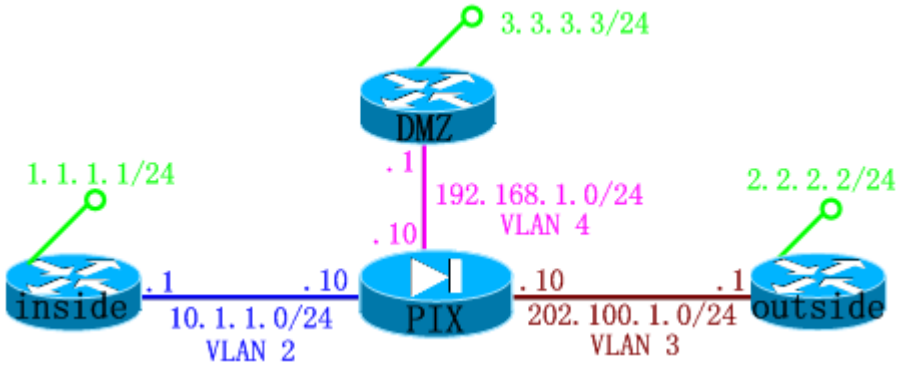
使用NAT命令,允许低安全级别访问高安全级别,必需要有访问列表:
PIX(config)# nat (outside) 1 202.100.1.0 255.255.255.0 outside
PIX(config)# global (inside) 1 interface

PIX(config)# clear configure nat
PIX(config)# clear configure global
PIX(config)# show run nat
PIX(config)# show run global

PIX(config)# clear xlate　　<=====P167<<安全PIX防火墙>>
PIX(config)# show xlate
0 in use, 4 most used

卷一实验

卷一实验：



Nat:

```
pixfirewall(config)# nat (inside) 1 10.1.1.0 255.255.255.0
pixfirewall(config)# global (outside) 1 202.100.1.100-202.100.1.200
R2_out>show user
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:05:46
* 66 vty 0                 idle                 00:00:00 202.100.1.100
  Interface    User               Mode         Idle     Peer Address
pixfirewall# show xlate
1 in use, 1 most used
Global 202.100.1.100 Local 10.1.1.1
pixfirewall# show conn
1 in use, 2 most used
TCP out 202.100.1.1:23 in 10.1.1.1:11007 idle 0:01:11 bytes 242 flags UIO
```

Pat:

```
pixfirewall(config)# nat (inside) 1 10.1.1.0 255.255.255.0
pixfirewall(config)# global (outside) 1 202.100.1.18
pixfirewall(config)# global (outside) 1 interface
pixfirewall# show xlate
1 in use, 2 most used
PAT Global 202.100.1.18(1025) Local 10.1.1.1(11020)
```

Nat(Static):

```
pixfirewall(config)# nat (inside) 1 10.1.1.1 255.255.255.255
pixfirewall(config)# global (outside) 1 202.100.1.101
pixfirewall(config)# show xlate
1 in use, 2 most used
```

PAT Global 202.100.1.101(1024) Local 10.1.1.1(11021)

Static:

```
pixfirewall(config)# static (inside,outside) 202.100.1.101 10.1.1.1
pixfirewall(config)# show xlate
1 in use, 2 most used
Global 202.100.1.101 Local 10.1.1.1
```

Port Redirect:

```
pixfirewall(config)# static (inside,outside) tcp interface 2323 10.1.1.1
23
pixfirewall(config)# show xlate
1 in use, 2 most used
PAT Global 202.100.1.10(2323) Local 10.1.1.1(23)
```

Port Redirect:

```
pixfirewall(config)# static (inside,outside) tcp interface 2323 10.1.1.1
23
pixfirewall(config)# access-list OUTACL extended permit tcp host
202.100.1.1 host 202.100.1.10 eq 2323
pixfirewall(config)# access-group OUTACL in interface outside
pixfirewall(config)# show xlate
1 in use, 2 most used
PAT Global 202.100.1.10(2323) Local 10.1.1.1(23)
R1_in>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:24:41
* 66 vty 0                 idle                 00:00:00 202.100.1.1
```

Static DOS defence:

```
pixfirewall(config)# static (inside,outside) 202.100.1.10 10.1.1.1 tcp
100 300 udp 500
```

tcp 100 ： 表示正常TCP连接最大的数量:100

tcp 300 ： 表示半开TCP连接最大的数量:300

udp 500 ： 表示UDP连接最大的数量:500

Note:英文半开连接有几种说法一种叫做"half-open",还有一种叫做"enbryonic"

Static DNS:

```
 pixfirewall(config)# static (inside,outside) 202.100.1.101 10.1.1.100
dns
```

Static Nat:

```
pixfirewall(config)# route outside 2.2.2.2 255.255.255.255 202.100.1.1
pixfirewall(config)# access-list G02 permit ip host 10.1.1.1 host 2.2.2.2
pixfirewall(config)# access-list G0202 permit ip host 10.1.1.1
202.100.1.0 255.255.255.0
pixfirewall(config)# static (inside,outside) 202.100.1.2 access-list G02
pixfirewall(config)# static (inside,outside) 202.100.1.202 access-list
G0202
```

Static Pat:

```
PIX(config)# access-list G2323 extended permit tcp host 10.1.1.1 eq 23
202.100.1.0 255.255.255.0
PIX(config)# access-list G23 extended permit tcp host 10.1.1.1 eq 23 host
2.2.2.2
PIX(config)# static (inside,outside) tcp 202.100.1.101 2323 access-list
G2323
PIX(config)# static (inside,outside) tcp interface 23 access-list G23
PIX(config)#access-list OUT extended permit tcp host 202.100.1.1 host
202.100.1.101 eq 2323
PIX(config)#access-list OUT extended permit tcp host 2.2.2.2 host
202.100.1.10 eq telnet
PIX(config)#access-group OUT in interface outside
R2_out>telnet 202.100.1.101 2323
R1_in#show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:29:43
* 66 vty 0                 idle                 00:00:00 202.100.1.1
  Interface    User               Mode         Idle     Peer Address
R1_in#show tcp brief
TCB       Local Address           Foreign Address         (state)
82F26884  10.1.1.1.23             202.100.1.1.11012       ESTAB
R2_out>telnet 202.100.1.10 /source-interface loopback 0
Trying 202.100.1.10 ... Open
R1_in#show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:32:06
* 66 vty 0                 idle                 00:00:00 2.2.2.2
  Interface    User               Mode         Idle     Peer Address
R1_in#show tcp brief
TCB       Local Address           Foreign Address         (state)
82F26D28  10.1.1.1.23             2.2.2.2.11013           ESTAB
```

Policy Nat(different target base):

```
PIX(config)# access-list NAT202 permit tcp host 10.1.1.1 host 202.100.1.1
PIX(config)# access-list NAT2 permit tcp host 10.1.1.1 host 2.2.2.2
PIX(config)# nat (inside) 1 access-list NAT202
PIX(config)# global (outside) 1 202.100.1.202
PIX(config)# nat (inside) 2 access-list NAT2
PIX(config)# global (outside) 2 202.100.1.2
R1_in>telnet 202.100.1.1
Trying 202.100.1.1 ... Open
R2_out>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:05:04
* 66 vty 0                 idle                 00:00:00 202.100.1.202
R2_out>show tcp brief
TCB       Local Address          Foreign Address        (state)
82F26798  202.100.1.1.23         202.100.1.202.1025     ESTAB
R1_in>telnet 2.2.2.2
Trying 2.2.2.2 ... Open
R2_out>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:06:41
* 66 vty 0                 idle                 00:00:00 202.100.1.2
  Interface    User               Mode         Idle     Peer Address
R2_out>show tcp brief
TCB       Local Address          Foreign Address        (state)
82F26C3C  2.2.2.2.23             202.100.1.2.1025       ESTAB
------------------------------------------------------------------------
---
Policy Nat(different target port base):
R2_out(config)#line vty 0 4
R2_out(config-line)#rotary 88
PIX(config)# access-list NAT88 permit tcp host 10.1.1.1 host 202.100.1.1
eq 3088
PIX(config)# access-list NAT23 permit tcp host 10.1.1.1 host 202.100.1.1
eq 23
PIX(config)# nat (inside) 1 access-list NAT88
PIX(config)# global (outside) 1 202.100.1.88
PIX(config)# nat (inside) 2 access-list NAT23
PIX(config)# global (outside) 2 202.100.1.23
R1_in>telnet 202.100.1.1
R2_out>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:00:22
* 66 vty 0                 idle                 00:00:00 202.100.1.23
R2_out>show tcp brief
```

```
TCB       Local Address           Foreign Address         (state)
82F26798  202.100.1.1.23          202.100.1.23.1024       ESTAB
R1_in>telnet 202.100.1.1 3088
R2_out>show users
    Line       User       Host(s)              Idle       Location
   0 con 0                 idle                 00:01:35
* 66 vty 0                 idle                 00:00:00 202.100.1.88
R2_out>show tcp brief
TCB       Local Address           Foreign Address         (state)
82F26C3C  202.100.1.1.3088        202.100.1.88.1024       ESTAB
```

Bypass Nat(identity nat):

```
PIX(config)# nat (inside) 0 10.1.1.0 255.255.255.0
```

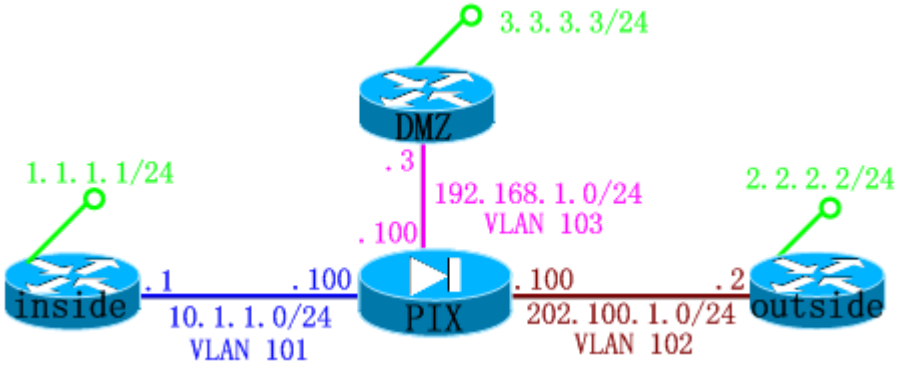
Bypass Nat(nat exemption):

```
PIX(config)# access-list NATEXEMPTION permit ip host 10.1.1.1 host
202.100.1.1
PIX(config)# nat (inside) 0 access-list NATEXEMPTION
R1_in#telnet 202.100.1.1
Trying 202.100.1.1 ... Open
R1_in#telnet 192.168.1.1
Trying 192.168.1.1 ...
% Connection refused by remote host
```

卷二实验

卷二实验:

Pre-share key L2L



1. 配置"outside"路由器,使"outside"路由器与"PIX"建立LAN-TO-LAN
2. 激活外部接口(IKE I)

   PIX(config)# crypto isakmp enable outside
3. 配置Pre-share KEY

   PIX(config)# crypto isakmp key cisco address 202.100.1.2

   或者: tunnel-group 202.100.1.2 type ipsec-121

```
tunnel-group 202.100.1.2 ipsec-attributes
pre-shared-key *
```

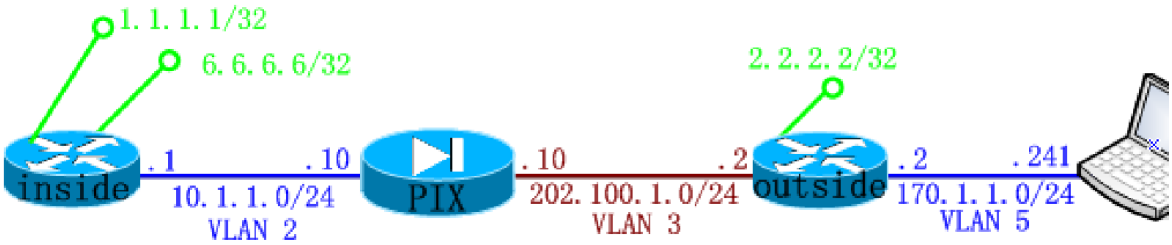
4. 激活外部接口(IKE II)

```
PIX(config)# crypto map huawei interface outside
```

5. 定义SYSOPT ----- 放行解密后流量穿越PIX,注意这个配置是默认配置,可以不敲

```
PIX(config)# sysopt connection permit-vpn
```

Pre-share key Remote



IKE I:

```
PIX(config)# isakmp policy 10 hash md5
PIX(config)# isakmp enable outside
```

IKE II:

```
PIX(config)# crypto ipsec transform-set wolf esp-des esp-md5-hmac
PIX(config)# crypto dynamic-map DMAP 10 set transform-set wolf
PIX(config)# crypto map SMAP 10 ipsec-isakmp dynamic DMAP
PIX(config)# crypto map SMAP interface outside
PIX(config)# ip local pool CCIE-POOL 192.168.168.100-192.168.168.200
```

定义tunnel-group:

```
PIX(config)# tunnel-group IPSECGROUP type ipsec-ra
PIX(config)# tunnel-group IPSECGROUP type ?
configure mode commands/options:
  ipsec-l2l   IPSec Site to Site group
  ipsec-ra    IPSec Remote Access group
PIX(config)# tunnel-group IPSECGROUP general-attributes
PIX(config-tunnel-general)# address-pool CCIE-POOL
PIX(config-tunnel-general)# tunnel-group IPSECGROUP ipsec-attributes
PIX(config-tunnel-ipsec)# tunnel-group IPSECGROUP ?
configure mode commands/options:
  general-attributes  Enter the general-attributes sub command mode
  ipsec-attributes    Enter the ipsec-attributes sub command mode
  ppp-attributes      Enter the ppp-attributes sub command mode
PIX(config-tunnel-ipsec)# pre-shared-key cisco123
PIX(config-tunnel-ipsec)# username huawei password huawei
```

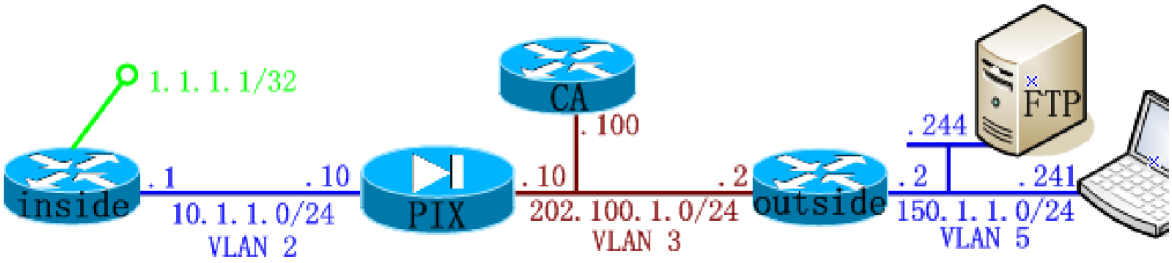
Tunnel split的配置:

```
PIX(config)# access-list SPLIT permit ip host 1.1.1.1 any
PIX(config)# group-policy SPLIT internal
PIX(config)# group-policy SPLIT attributes
PIX(config-group-policy)# split-tunnel-policy tunnelspecified
PIX(config-group-policy)# split-tunnel-network-list value SPLIT
```

用户调用group-policy策略:

```
PIX(config)# username huawei attributes
PIX(config-username)# vpn-group-policy SPLIT
```

RSA-sig Remote VPN



卷五实验

卷五实验:

Transparent Mode:

Third layer stream through

R1和R3起OSPF,并且宣告

```
PIX# show firewall
Firewall mode: Router
PIX# show mode
Security context mode: single
PIX(config)# firewall transparent
PIX(config)# int e0
PIX(config-if)# nameif outside
PIX(config-if)# no sh
PIX(config-if)# int e1
PIX(config-if)# nameif inside
PIX(config-if)# no sh
PIX(config)# ip add 13.1.1.100 255.255.255.0
```

到此为止:R1与R3能互发HELLO包,但不能互收到DPD包.

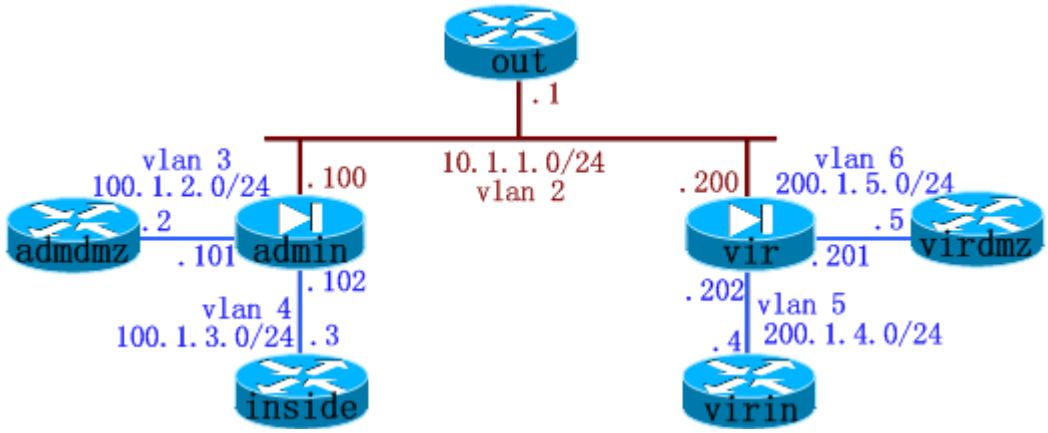
```
R3#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
Interface
1.1.1.1           1   EXSTART/BDR     00:00:37    13.1.1.1
Ethernet0/0
PIX(config)# access-list ACLIN permit ospf any any
PIX(config)# access-group ACLIN in interface inside
PIX(config)# access-list ACLOUT permit ospf any any
```

```
PIX(config)# access-group ACLOUT in interface outside
到此为止:R1与R3能互相收到路由,但PING不通
R3#show ip route
     1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/11] via 13.1.1.1, 00:00:05, Ethernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
C       3.3.3.3 is directly connected, Loopback0
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, Ethernet0/0
PIX(config)# access-list ACLIN permit icmp any any
PIX(config)# access-list ACLOUT permit icmp any any
```

证明: 1. Transparent firewall 默认只对进来的包做policy ,默认对出去的包放行.

2. Transparent firewall 默认只对从outside 接口进来的包做检查,对从inside接口进来的包不做检查.

Multi context Mode:



基本配置:

```
pixfirewall(config)# mode multiple
WARNING: This command will change the behavior of the device
WARNING: This command will initiate a Reboot
pixfirewall(config)# interface ethernet 0
pixfirewall(config-if)# no shutdown
pixfirewall(config)# interface ethernet 1
pixfirewall(config-if)# no shutdown
pixfirewall(config-if)# int ethernet 1.3
pixfirewall(config-subif)# vlan 3
pixfirewall(config-subif)# interface ethernet 1.4
pixfirewall(config-subif)# vlan 4
pixfirewall(config-subif)# interface ethernet 1.5
pixfirewall(config-subif)# vlan 5
pixfirewall(config-subif)# interface ethernet 1.6
```

```
pixfirewall(config-subif)# vlan 6
```

创建防火墙:

```
pixfirewall(config)# admin-context admin
pixfirewall(config)# context Vir
pixfirewall(config-ctx)# config-url flash:/vir.cfg
```

对相应的防火墙关连相应的接口:

```
pixfirewall(config-ctx)# context admin
pixfirewall(config-ctx)# allocate-interface e1.3-e1.4
pixfirewall(config-ctx)# allocate-interface e0
pixfirewall(config)# context Vir
pixfirewall(config-ctx)# allocate-interface e1.5-e1.6
pixfirewall(config-ctx)# allocate-interface e0
```

Enter virtual firewall:

```
pixfirewall(config)# changeto context admin
pixfirewall/admin(config)# interface ethernet 0
pixfirewall/admin(config-if)# ip add 10.1.1.100 255.255.255.0
pixfirewall/admin(config-if)# nameif outside
pixfirewall/admin(config-if)# interface ethernet 1.3
pixfirewall/admin(config-if)# ip add 100.1.2.101 255.255.255.0
pixfirewall/admin(config-if)# nameif dmz
pixfirewall/admin(config-if)# security-level 50
pixfirewall/admin(config-if)# int ethernet 1.4
pixfirewall/admin(config-if)# nameif inside
pixfirewall/admin(config-if)# ip add 100.1.3.102 255.255.255.0
```

存盘:

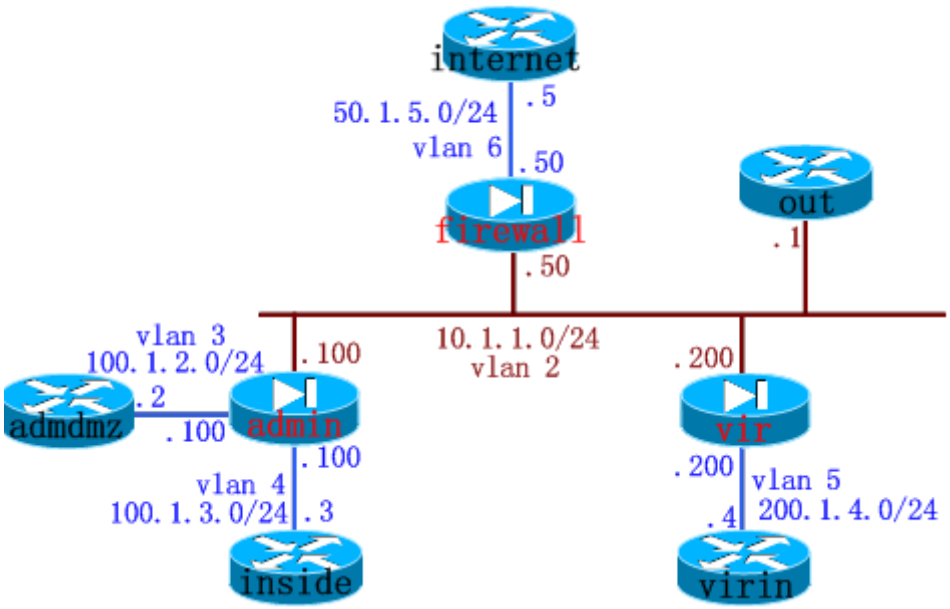
```
pixfirewall/admin# write
pixfirewall/Vir# write
pixfirewall/admin# changeto system
pixfirewall# write
```

解决相同接口的问题:

假设上面的"virtual firewall'admin"与"virtual firewall'Vir",使用了相同接口,可使用更改MAC地址:

```
pixfirewall/admin(config-if)# mac-address aaaa.1111.1111
```

堆叠的Multi context 配置实例:




```
pixfirewall/admin# show conn
pixfirewall/admin# show xlate
```

Failover

```
failover lan enable
failover lan unit primary
failover lan interface failover Ethernet1
failover interface ip failover 201.1.254.10 255.255.255.0 standby
201.1.254.20
failover key cisco
failover
```

VPN配置

PIX525-IPSEC-VPN配置
实验要求：
在PIX525上进行配置，要求内网PC2能telnet 登入到PIX525上；外网PC1可以通过SSH方式登入到
PIX525；通过在PIN525上配置NAT 使内网PC2可以ping通外网的R1，R2，PC1；在PIX525上配置
IPSEC-VPN，使PC1可以通过IPSEC-VPN 方式登陆到内网上，并PING通内网主机PC2。
IP地址表：（如图）

pc1<----------------------->R1<----------------------->R2
<----------------------->PIX<----------------------->pc2

E0 S0 S0
E0 E0 E1

11.1.1.10/24 12.1.1.1/24 23.1.1.2/24
192.168.11.254/24

11.1.1.254/24 12.1.1.2/24
23.1.1.3/24 192.168.11.10/24
实验配置如下:
一、路由器和PC 机配置
配置R1：

```
Router>en
Router#conf t
Router(config)#hostname R1
R1(config)#int S0
R1(config-if)#ip add 12.1.1.1 255.255.255.0
R1(config-if)#no shut

R1(config-if)#int E0
R1(config-if)#ip add 11.1.1.254 255.255.255.0
R1(config-if)#no shut
R1(config-if)#router rip
R1(config-router)#network 12.1.1.1
R1(config-router)#network 11.1.1.0
R1(config-router)#end
R1#sh run
```

配置R2：

```
Router>en
Router#conf t
Router(config)#hostname R2
R2(config)#int S0
R2(config-if)#ip add 12.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#clock rate 64000
R2(config-if)#int E0
```

R2(config-if)#ip add 23.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#router rip
R2(config-router)#network 12.1.1.2
R2(config-router)#network 23.1.1.2
R2(config-router)#end
R2#sh run
配置PC机：
PC1：修改本地连接的TCP/IP属性，设置IP：11.1.1.10 掩码：255.255.255.0 网关：11.1.1.254
PC2：修改本地连接的TCP/IP属性，设置IP：192.168.11.10 掩码：255.255.255.0 网关：192.168.11.254
二、PIX525防火墙配置：
1.接口配置：
pixfirewall>en
pixfirewall#conf t
pixfirewall(config)#ho PIX525
PIX525(config)#int E0
PIX525(config-if)#ip add 23.1.1.3 255.255.255.0

PIX525(config-if)#nameif outside //设置接口为外网接口，启动安全级别为0
PIX525(config-if)#security-level 0
PIX525(config-if)#no shut
PIX525(config-if)#int E1
PIX525(config-if)#ip add 192.168.11.254 255.255.255.0
PIX525(config-if)#nameif inside //设置接口为内网接口，启动安全级别为100
PIX525(config-if)#security-level 100
PIX525(config-if)#no shut
PIX525(config-if)#exit
PIX525(config)#
2配置允许内网telnet登入：
PIX525(config)#telnet 0.0.0.0 0.0.0.0 inside //设置E1连接的内网所有主机可以远程登入到PIX上
PIX525(config)#passwd spoto //设置内网主机Telnet 登入到PIX上的密码
PIX525(config)#enable password spoto
3.配置允许外网SSH 登入：
PIX525(config)#ca zeroize //清除原有的CA配置
PIX525(config)#ca generate //配置CA为普通级别配置
PIX525(config)#ca save //保存CA配置
PIX525(config)#ssh 0.0.0.0 0.0.0.0 outside //允许外部所有网络通过SSH 方式从E0口登入
PIX525(config)#username admin password admin //建立一本地用户，VPN和SSH 登入时使用
PIX525(config)#aaa authentication ssh LOCAL //使用本地用户认证
4.配置PAT：
PIX525(config)#global (outside) 1 interface //通过出接口方式转换为外网地址
PIX525(config)#nat (inside) 1 192.168.11.0 255.255.255.0 //允许转换的内部地址网络

PIX525(config)#route outside 0.0.0.0 0.0.0.0 23.1.1.2 //起默认路由，让内网能ping通外网网段
为了让内网的机器能够PING 出去还要加上： //默认情况下，外网是不允许ping 通内网的，当内网的
ping 包出去后，在返回时候会被outside 接口拒绝或者丢弃，所以要让outside 接口能接受这个包，作
如下配置：
PIX525(config)#access-list 100 permit icmp any any
PIX525(config)#access-group 100 in interface outside
5.IPSEC-VPN配置：
PIX525(config)#ip local pool ipsecvpn 192.168.11.20-192.168.11.120 mask 255.255.255.0
//建立VPN 动态IP 地址池,当外网用户使用IPSEC--VPN方式登陆时候，自动从IP池分配一个IP 给用户
PIX525(config)#access-list nonat permit ip 192.168.11.0 255.255.255.0 192.168.11.0 255.255.255.0
//建立列表允许内网网段（含VPN 网段）通过PIX 防火墙时，不需要NAT 转换

PIX525(config)#nat (inside) 0 access-list nonat //应用访问控制列表，0表示不进行转换
PIX525(config)# sysopt connection permit-vpn
－－－－－－
PIX525(config)#access-list tunnellist permit 192.168.11.0 255.255.255.0
PIX525(config)#group-policy spoto internal //建立内部组策略，命名为spoto
PIX525(config)#group-policy spoto attributes
PIX525(config-group-policy)#vpn-idle-timeout 20 //设置VPN超时时间为20钟
PIX525(config-group-policy)#split-tunnel-policy tunnelspecified //建立隧道分离策略
PIX525(config-group-policy)#split-tunnel-network-list value tunnellist
//与tunnellist匹配的网络将全部使用隧道分离
－－－－－－
PIX525(config-group-policy)#exit
PIX525(config)#crypto ipsec transform-set myset esp-des esp-md5-hmac
//ipsec的数据转换格式集通过des方式加密，对方通过哈希表-md5方式还原数据
PIX525(config)#crypto dynamic map mydynmap 20 set transform-set myset
//建立加密动态映射图，并采用上建的myset 方式加密解密
PIX525(config)#crypto map mymap 20 ipsec-isakmp dynamic mydynmap
//建立加密静态映射图，加密静态映射图中ipsec-isakmp 采用上建的加密动态映射图加密
PIX525(config)#crypto map mymap interface outside //将加密静态映射图应用于外网接口
PIX525(config)#crypto isakmp identity address //isakmp采用地址验证
PIX525(config)#crypto isakmp enable outside //isakmp 应用于外网接口
// isakmp:Internet Security Association and Key Management Protocol policy.
互连网安全密钥管理协议策略 应用到外网接口
－－－－－－
PIX525(config)#crypto isakmp policy 10 //建立isakmp 策略
PIX525(config-isakmp-policy)#authentication pre-share //使用共享密钥认证
PIX525(config-isakmp-policy)#encryption des //使用des算法加密

PIX525(config-isakmp-policy)#hash md5 //哈希使用MD5算法还原数据
PIX525(config-isakmp-policy)#end

PIX525(config)#tunnel-group spoto10 type ipsec-ra //建立VPN 远程登入(即使用隧道分离)组
PIX525(config)#tunnel-group general-attributes //配置VPN远程登入(即使用隧道分离)组的基本属性
PIX525(config-tunnel-general)#address-pool ipsevvpn //设置VPN登入内网时分配的IP地址池
PIX525(config-tunnel-general)#authentication-server-group LOCAL //服务端组使用本地认证
－－－－－－
PIX525(config-tunnel-general)#default-group-policy spoto //指定默认的组策略为spoto
－－－－－－
PIX525(config-tunnel-general)#exit
PIX525(config)#tunnel-group spoto10 ipsec-attributes //设置VPN 远程登入(即使用隧道分离)组的
ipsec属性
PIX525(config-tunnel-ipsec)#pre-share-key spoto //设置使用的共享密钥为spoto
6.验证：
在外网PC1 上使用VPN 方式测试是否能成功登入到PIX525 上（需要安装VPN-CLIENT 软件），在
VPN-CLIENT 上host项上输入PIX525的outside接口地址，然后输入VPN 组spoto10和密码spoto，
点击连接，等待输入用户名和密码。输入完用户名密码等待PIX验证后，在运行下输入cmd进入CLI模
式。在CLI模式下输入ipconfig /all查看是否获取到ipsecvpn 地址池中分配到的IP 地址，若分配到则
成功使用VPN方式登入到了PIX 上，确切说登入到了内部网络上。这样PC1就如同在内部网络中一样可
以随意使用内部网络资源了！

安装ASDM

让我们开始吧。下面是我们需要发布的指令和让ASDM运行的步骤:

1.登录到PIX并且进入启用模式:“pix> enable”

2.进入启用模式之后，输入命令“copy tftp flash”，你现在可以看到弹出如下信息:

3.“Address or name of remote host [x.x.x.x]? ”。你需要在这里输入托管ASDM图像的TFTP服务器的IP地址。按回车键继续操作。

4.“Source file name [cdisk]? ”。输入ASDM图像的文件名，例如:asdm502.bin for ASDM version 5.0(2)。按回车键继续操作。

5.“Destination file name [asdm502.bin]?”。这里实际上没有任何事情可做，除非你要重新命名你正在传输的图像。所以，这里按回车键。

6.我们需要告诉PIX软件ASDM在什么地方。因此，我们要要以配置模式发出下列命令。如果你喜欢这种长的方式，你可以在CLI输入“conf t”或者“configure terminal”。一旦进入设置模式“pix(config)#”,然后输入“asdm image flash:asdm502.bin”，然后按回车键。

7.由于我们已经让我们的PIX知道了ASDM在什么地方，现在可以向PIX发出“write mem”或者“write memory”指令。你将得到一个信息，说正在构建配置，然后将返回到“pix(config)#”。这个时候，我们就安装完了ASDM。

为了访问ASDM，我们需要做几件事情。否则，这个PIX软件将拒绝通信并且切断连接。为了允许这个连接，我们需要以config(配置)模式发布如下指令:

• http server enable:这个指令要首先发布并且启动http/https服务器。

• http 0 0 inside:这个指令能够启动来自PIX内部设置的任何主机/网络的通信。如果你仅允许你的工作站通信，这个工作站的地址是192.168.89.44，那么，这个指令就是“http 192.168.89.44 255.255.255.255 inside”。你还可以允许一个子网或者多个子网连接。如果你需要在任何时候撤销任何入口，简单地使用这个指令就可以了“no http x.x.x.x z.z.z.z inside”。这里的x代表IP地址，z代表子网。

现在，你可以实验一下，使用https://x.x.x.x/admin连接ASDM。在这里，x代表在PIX接口内部的IP地址。

请注意，从接口外部也可以访问ASDM。你需要确认，当你添加“http x.x.x.x z.z.z.z ”指令的时候，你是在指定这个接口为外部接口，用一台安全的计算机可以访问那个接口。然而，这种方法不推荐使用，由于ASDM的强大功能，把ASDM放在可以公开访问的网络上是不明智的。

ASDM应该安装完毕并且可以工作了。使用你的PIX登录启用口令，这个软件就开始快速运行了，除非你遇到了问题。在这个讲座的下半部分，我们将研究ASDM的故障排除问题。我将提供输出样本供你们参考。

IPS

Initial/SPAN/RSPAN

为防止骇客通过其中一台被攻陷的内网服务器去访问另一台内网服务器:
Switch(config-if)#switchport protected <===端口隔离(在连接到所有服务器上的端口都敲上此命令,但连接防火墙的端口不敲)

show run <===> more current-config <===> show configuration

```
sensor#erase current-config
sensor#show interfaces
MAC statistics from interface FastEthernet0/1
   Interface function = Sensing interface
MAC statistics from interface FastEthernet0/0
   Interface function = Command-control interface
```

IPS的初始化: P137

清空配置/SETUP/配置SSH的Server(边界路由器)/向SSH的Server获取密码/添加和删除用户

SSH的Server的作用:

假设IPS要登录到"SSH的Server"上,写"access-list",做BLOCK.那就要向"SSH的Server"获取密码.

假设"SSH的Server"有多台,那就要向每一台"SSH的Server"获取密码.

SPAN ---- Switch Port Analyzer : P100

基于VLAN的抓包:只能抓进入的包,不能抓出去的包 <===35系列以上支持(基于VLAN叫VSPAN)

基于端口的抓包:双方向

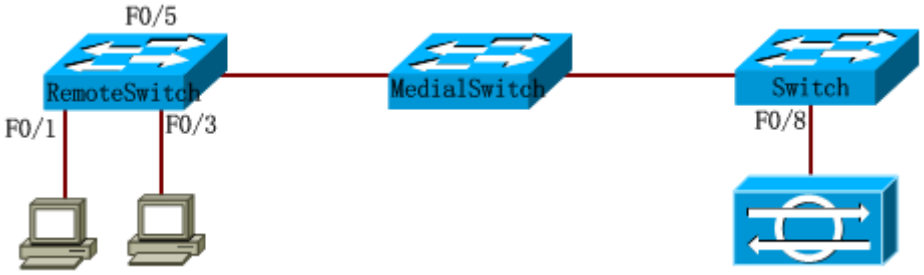
配置: P103

需要抓取的流量在多个VLAN里面时:

```
Switch(config)#monitor session 1 destination interface fastethernet 0/12
encapsulation dot1q <==可以选dot1q/ISL(但:IDS不支持ISL); 能抓任何VLAN的包

Switch(config)#monitor session 1 destination interface fastethernet 0/12
ingree vlan 1    <==只能往VLAN 1发reset包

Switch(config)#monitor session 1 destination interface fastethernet 0/12
encapsulation dot1q ingree vlan 1    <====能往任何VLAN发reset包(只根据:"encapsulation dot1q" ,不根据:"ingree vlan 1" )

Switch(config)#monitor session 2 source vlan 11 rx      <===抓取VLAN 11进入的包(3550系列以上才支持两个session)
```

RSPAN --- Remote Switch Port Analyzer : P105

实验 1 :



第一步:

```
RemoteSwitch(config)#vlan 100
RemoteSwitch(config-vlan)#remote-span
MedialSwitch(config)#vlan 100
MedialSwitch(config-vlan)#remote-span
Switch(config)#vlan 100
Switch(config-vlan)#remote-span
```

第二步:正常抓取流量

第三步:

3550系列:

```
RemoteSwitch(config)#monitor session 1 destination remote vlan 100
reflector-port fastethernet 0/5
```

此接口必须是物理接口,且没接设备(空闲)

3560系列:

```
RemoteSwitch(config)#monitor session 1 destination remote vlan 100
```

第四步:

```
Switch(config)#monitor session 1 source remote vlan 100
```

第五步:

```
RemoteSwitch(config)#monitor session 1 destination  interface
fastethernet 0/8
```

实验 2 : 混合SPAN

需求:抓取 A --- B 之间的流量,并抓取 C --- D 之间的流量

.....

第四步:

```
Switch(config)#monitor session 1 source vlan 11 , 100 rx <===vlan 11是抓
```

取 C/D之间的流量(35系列以上)

第五步:

RemoteSwitch(config)#monitor session 1 destination interface fastethernet 0/8 encapsulation dot1q

IDM

```
R1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  12.1.1.1               -   0010.7b0e.3f61  ARPA   Ethernet0/0
Internet  12.1.1.2               6   0010.7b79.cb61  ARPA   Ethernet0/0

Switch#show mac-address-table
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0010.7b0e.3f61    DYNAMIC     Fa0/14
   1    0010.7b79.cb61    DYNAMIC     Fa0/9
Total Mac Addresses for this criterion: 02
```

1. IDS的初始化配置:P209

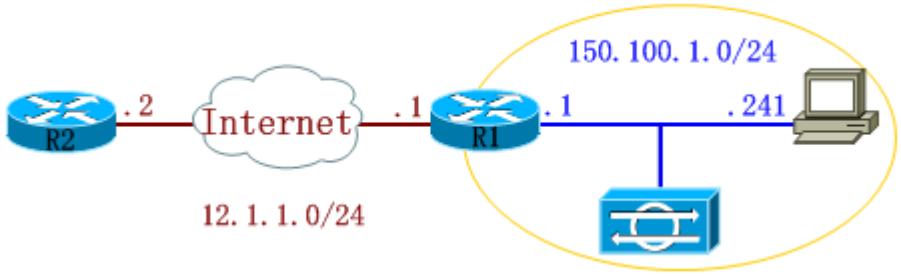
2. 配置允许的主机段:P212

3. 安全SSH属性(定义授权密钥/生成新的主机密钥/配置SSH已知主机密钥):P213

4. 证书管理(信任主机证书/产生主机证书/查看服务器证书):P218

5. IEV的使用: P173

实验 1 : 当PC访问R1时,URL中出现"cmd.exe"(例:"http://x.x.x.x/cmd.exe"),将会被reset



步骤: P269

1. 打开IDS ===> "Signature Wizard"

2. "Web Server Signature" :

Web Server Signature : <=== 基于端口
Packet Signatures : <====基于包(例:telnet的包)
Stream Signatures : <=== 基于流(例:抓取"root"字符串)

2. "Signature Identification" :
Signature ID : 任意(大于等于20000)

3. "Web Server Service Ports" :
Service Ports : 定义Http的端口,定义多个时可用逗号隔开(例:80,8180,8080)

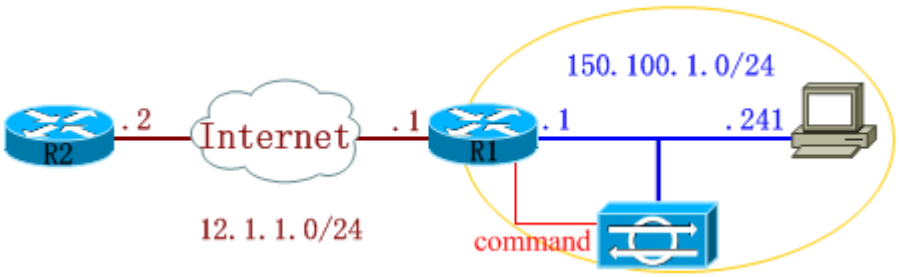
4. "Web Server Buffer Overflow Checks" <===用于防止缓存溢出

5. "Web Server Regular Expressions" :
HTTP Header URL Regular Expression : cmd\.exe

6. "Alert Response Actions" :
Include Packet in Alert : <===是否开启抓包

7. 打开 HTTP 服务器
R1(config)#ip http server
R1(config)#ip http authentication local
R1(config)#username cisco privilege 15 password cisco

实验 2 : R2路由器telnet到R1路由器的3038端口,将会被BLOCK掉



配置:

```
R1(config)#username cisco privilege 15 password cisco
R1(config)#line vty 0 4
R1(config-line)#rotary 38        <===telnet端口号:3000+38=3038
R1(config-line)#login local
----------------------------------
R2#telnet 12.1.1.1 3038
Trying 12.1.1.1, 3038 ... Open
User Access Verification
Username: cisco
Password:
----------------------------------
```

写BLOCK参数: P288 <=====路由器一定要存盘和给足够的权限给IDS写列表进路由器

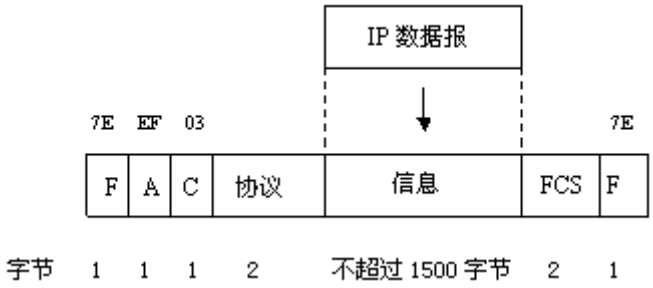
```
R2(config)#access-list 102 deny tcp any any established
R2(config)#access-list 102 permit tcp any any
```

tcpip

PPP

PPP帧格式和HDLC帧格式相似，如图所示。二者主要区别：PPP是面向字符的，而

HDLC是面向位的



可以看出，PPP帧的前3个字段和最后两个字段与HDLC的格式是一样的。标志字段F为0x7E（0x表示7E），但地址字段A和控制字段C都是固定不变的，分别为0xFF、0x03。PPP协议不是面向比特的，因而所有的PPP帧长度都是整数个字节。
与HDLC不同的是多了2个字节的协议字段。协议字段不同，后面的信息字段类型就不同。如：
0x0021--信息字段是IP数据报
0xC021--信息字段是链路控制数据LCP
0x8021--信息字段是网络控制数据NCP
0xC023--信息字段是安全性认证PAP
0xC025--信息字段是LQR
0xC223--信息字段是安全性认证CHAP
当信息字段中出现和标志字段一样的比特0x7E时，就必须采取一些措施。因PPP协议是面向字符型的，所以它不能采用HDLC所使用的零比特插入法，而是使用一种特殊的字符填充。具体的做法是将信息字段中出现的每一个0x7E字节转变成2字节序列（0x7D，0x5E）。若信息字段中出现一个0x7D的字节，则将其转变成2字节序列（0x7D，0x5D）。若信息字段中出现ASCII码的控制字符，则在该字符前面要加入一个0x7D字节。这样做的目的是防止这些表面上的ASCII码控制字符被错误地解释为控制字符。

MTU

```
R1#show int e0/0
Ethernet0/0 is administratively down, line protocol is down
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,          <===二层MTU

R1#show ip interface e0/0
  MTU is 1500 bytes                    <===三层MTU(针对IP协议)

R1(config-if)#mtu 1388                 <===二层MTU不允许修改
% Interface Ethernet0/0 does not support user settable mtu.

R1(config-if)#ip mtu 1288              <====三层MTU可以修改
```

ip

当R5允许远程telnet时:

```
R5#show tcp brief
TCB        Local Address               Foreign Address         (state)
62C999C8   56.1.1.5.23                 56.1.1.6.61658           ESTAB    <===对方已经发起telnet到我方(有没有验证密码没关系)
R5#show tcp brief
TCB        Local Address               Foreign Address         (state)
62C999C8   56.1.1.5.23                 56.1.1.6.61658          TIMEWAIT     <===等待对方发起telnet
R5#clear tcp tcb 62C999C8       <===强行清除TCP连接
[confirm]
[OK]
```

默认源目IP地址是不会被改变的(除:"NAT"会改变源,"源站路由"会改变目的)
MAC地址是每一跳都会改变

关闭源站选路:Router(config)#no ip source-route　　<===CISCO路由器默认打开源站选路

在路由器上将IP地址与MAC地址静态绑定:

```
Router(config)#arp 1.1.1.1 1111.2222.3333 arpa
Router#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  1.1.1.1                -    1111.2222.3333  ARPA
```

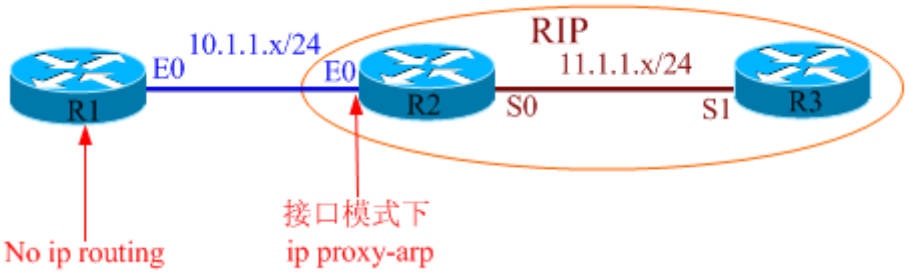
ARP包的工作原理:

步骤:

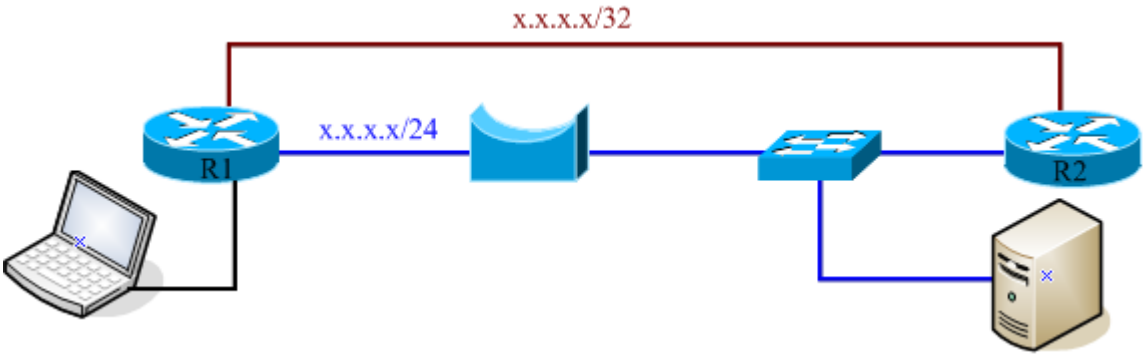
1. PC1发送ARP请求
2. 交换机CAM表中记录:"PC1的MAC地址和接口的绑定"
3. 在PC2上:将PC1的MAC地址写到PC2的ARP表中
4. R2回应ARP请求
5. 交换机CAM表中记录:"PC2的MAC地址和接口的绑定"

ARP代理

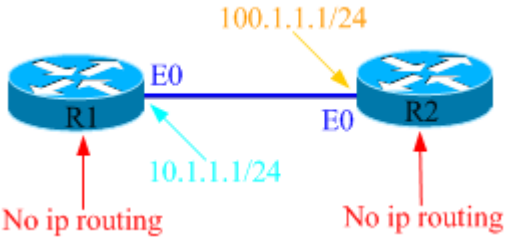
实验 1:　接口下的ARP代理默认为开启状态



实验 2:



实验 3:



免费ARP

Switch(config-if)#switchport protected <====所有输入此命令的端口都不能互相PING通,但和没有输入此命令的端口能互相PING通

ICMP

ICMP Format

|<--IP首部-->|<---------- ICMP报文 ---------->|

...| IP首部 | ICMP | IP首部 | ICMP差错报文的IP数据报的前8个字节 | ...

s:R2 d:R1 3 0 s:R1 d:R3

Traceroute

实验 1:

```
R2#debug ip icmp
R2#traceroute 13.1.1.3
Type escape sequence to abort.
Tracing the route to 13.1.1.3
00:10:33: ICMP: time exceeded rcvd from 12.1.1.1
00:10:33: ICMP: time exceeded rcvd from 12.1.1.1
00:10:33: ICMP: time exceeded rcvd from 12.1.1.1
00:10:33: ICMP: dst (12.1.1.2) port unreachable rcv from 13.1.1.3
00:10:36: ICMP: dst (12.1.1.2) port unreachable rcv from 13.1.1.3
```

```
R2#debug ip packet detail
R2#traceroute 13.1.1.3
Type escape sequence to abort.
Tracing the route to 13.1.1.3
  1 12.1.1.1 32 msec 44 msec 32 msec
  2 13.1.1.3 44 msec
00:18:28: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:28:     UDP src=35161, dst=33434
00:18:28: IP: s=12.1.1.1 (Serial0), d=12.1.1.2 (Serial0), len 56, rcvd 3
00:18:28:     ICMP type=11, code=0
00:18:28: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:28:     UDP src=38415, dst=33435
00:18:28: IP: s=12.1.1.1 (Serial0), d=12.1.1.2 (Serial0), len 56, rcvd 3
00:18:28:     ICMP type=11, code=0
00:18:28: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:28:     UDP src=33959, dst=33436
00:18:28: IP: s=12.1.1.1 (Serial0), d=12.1.1.2 (Serial0), len 56, rcvd 3
00:18:28:     ICMP type=11, code=0
00:18:29: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:29:     UDP src=32867, dst=33437
00:18:29: IP: s=13.1.1.3 (Serial0), d=12.1.1.2 (Serial0), len 56, rcvd 3
00:18:29:     ICMP type=3, code=3
00:18:29: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:29:     UDP src=33156, dst=33438 *  44 msec
00:18:32: IP: s=12.1.1.2 (local), d=13.1.1.3 (Serial0), len 28, sending
00:18:32:     UDP src=33504, dst=33439
00:18:32: IP: s=13.1.1.3 (Serial0), d=12.1.1.2 (Serial0), len 56, rcvd 3
00:18:32:     ICMP type=3, code=3
```

```
------------------------------------------------------------------------------
--
R2#show debugging
Generic IP:
  ICMP packet debugging is on
------------------------------------------------------------
```

实验 2:


```
R1#ping
Target IP address: 3.3.3.3
Extended commands [n]: y
Loose, Strict, Record, Timestamp, Verbose[none]: strict
Source route: 12.1.1.2 23.1.1.3
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet has IP options:  Total option bytes= 11, padded length=12
 Strict source route: <*>
   (12.1.1.2)
   (23.1.1.3)
Reply to request 0 (16 ms).  Received packet has options
 Total option bytes= 12, padded length=12
 Strict source route:
   (23.1.1.3)
   (12.1.1.2)
   <*>
 End of list
.....
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/16 ms
------------------------------------------------------------------------------
----
R3#debug ip icmp
ICMP packet debugging is on
R3#
*Mar  1 02:00:33.939: ICMP: echo reply sent, src 3.3.3.3, dst 12.1.1.1
*Mar  1 02:00:33.951: ICMP: echo reply sent, src 3.3.3.3, dst 12.1.1.1
*Mar  1 02:00:33.963: ICMP: echo reply sent, src 3.3.3.3, dst 12.1.1.1
*Mar  1 02:00:33.975: ICMP: echo reply sent, src 3.3.3.3, dst 12.1.1.1
*Mar  1 02:00:33.987: ICMP: echo reply sent, src 3.3.3.3, dst 12.1.1.1
------------------------------------------------------------------------------
R1#traceroute
Target IP address: 12.1.1.1
```

```
Source address: 12.1.1.1
Loose, Strict, Record, Timestamp, Verbose[none]: 1
Source route: 23.1.1.3
Tracing the route to 12.1.1.1
  1 12.1.1.2 4 msec
  2 23.1.1.3 8 msec
  3 23.1.1.2 8 msec
  4 12.1.1.1 8 msec
```

IP选路

```
C:\>netstat -r            <===查看路由表(或者:  C:\>route print)

Route Table
===========================================================================
==
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 17 31 ad ec c3 ...... Realtek RTL8139 Family PCI Fast Ethernet
NIC
数据包计划程序微型端口
0x3 ...00 03 c9 7b f0 7a ...... Intel(R) PRO/Wireless 2200BG Network
connect
- 数据包计划程序微型端口
===========================================================================
==
===========================================================================
==
Active Routes:
Network Destination        Netmask          Gateway       Interface
Metric
          0.0.0.0          0.0.0.0     192.168.16.1  192.168.16.128
1
        127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1
1
     192.168.16.0    255.255.255.0  192.168.16.128  192.168.16.128
1
   192.168.16.128  255.255.255.255        127.0.0.1        127.0.0.1
1
   192.168.16.255  255.255.255.255  192.168.16.128  192.168.16.128
1
        224.0.0.0        240.0.0.0  192.168.16.128  192.168.16.128
1
```

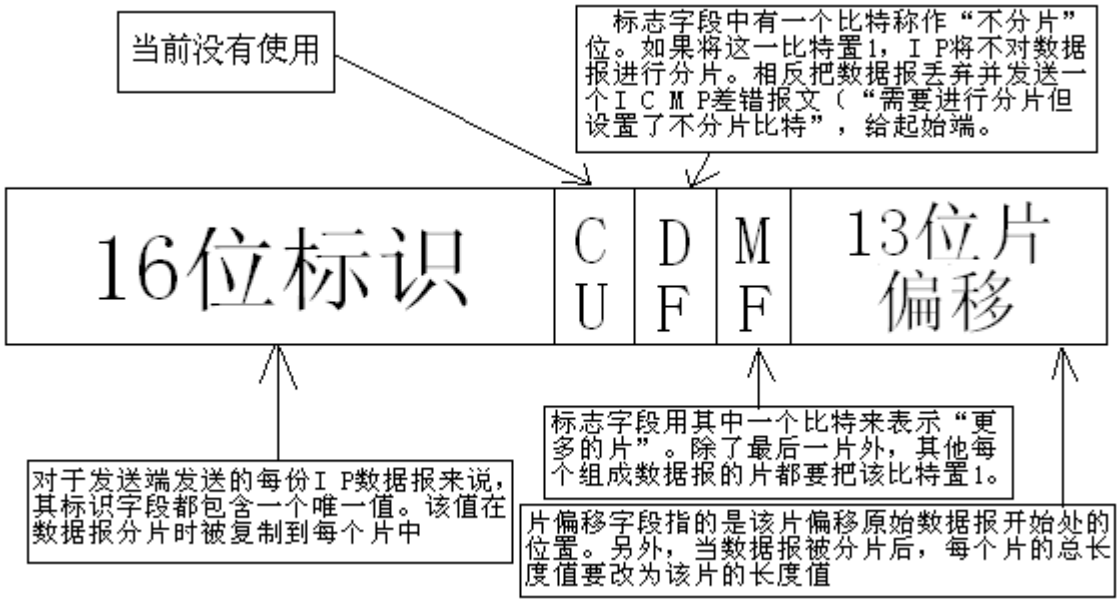
```
  255.255.255.255   255.255.255.255    192.168.16.128                2
1
  255.255.255.255   255.255.255.255    192.168.16.128   192.168.16.128
1
Default Gateway:       192.168.16.1
===========================================================================
==
Persistent Routes:
  None
```

CISCO 的选路策略:

1. 明细策略路由 (router map set next hop)
2. 明细路由 (静态路由/动态路由)
3. 策略默认路由 (router map set default)
4. 默认路由 (0.0.0.0)

```
R1(config-if)#no ip redirects        <====关闭重定向功能
```

IP 分片



偏移量:第一个包的偏移量为 : "0"
第二个包的偏移量为 : "第一个包的字节数"
第三个包的偏移量为 : "第一个包的字节数+第二个包的字节数"

fragment acl 对包的处理:

ACE no-fragment
1. no-initial fragment .per 通过
   .deny 处理下一条
2. nofragment and initial fragment .正常处理

ACE fragment　3层信息　1. no-initial fragment　正常处理
2. nofragment and initial fragment　处理下一条

4层信息　permit 处理三层如果OK就PASS
deny　drop it

ACE是访问控制列表中的一个条目.offset为"零",就不是分片包；offset为"大于零",就是分片包.

```
R1(config)#access-list 101 permit ip host 1.1.1.1 host 2.2.2.2
R1(config)#access-list 101 permit ip host 1.1.1.1 host 2.2.2.2 fragments
R1(config)#access-list 101 permit tcp host 1.1.1.1 host 2.2.2.2 fragments
```

ESTABLISHED的作用:　("established"命令只检查:"ack" "rst" "fin")
以下命令的作用:只允许内部向外部建立连接,不允许外部向内部建立连接

```
R1(config)#access-list 101 permit tcp any any established
R1(config)#access-list 101 deny ip any any
```

UDP

```
C:\>netstat -an
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
```

TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1025	0.0.0.0:0	LISTENING
TCP	192.168.16.128:139	0.0.0.0:0	LISTENING
TCP	192.168.16.128:1296	192.168.16.170:23	ESTABLISHED
UDP	0.0.0.0:445	*:*	
UDP	0.0.0.0:500	*:*	
UDP	0.0.0.0:1031	*:*	
UDP	0.0.0.0:1035	*:*	
UDP	0.0.0.0:4500	*:*	
UDP	127.0.0.1:123	*:*	
UDP	127.0.0.1:1298	*:*	
UDP	127.0.0.1:1300	*:*	
UDP	127.0.0.1:1900	*:*	
UDP	127.0.0.1:10000	*:*	
UDP	192.168.16.128:123	*:*	
UDP	192.168.16.128:137	*:*	
UDP	192.168.16.128:138	*:*	
UDP	192.168.16.128:1900	*:*	

TCP

TCP ECHO

```
R1(config)#service tcp-small-servers          <===打开TCP ECHO(默认关闭)
R1#telnet 12.1.1.1 echo
Trying 12.1.1.2, 7 ... Open
aabbcc

R1#telnet 12.1.1.1 daytime
Trying 12.1.1.1, 13 ... Open
Wednesday, May 30, 2007 16:11:12-GTM
[Connection to 12.1.1.1 closed by foreign host]

R1#telnet 12.1.1.1 chargen       <====检测线路的好坏(代替PING)

R1#telnet 12.1.1.1 discard       <=====输入此命令后之后,输入所有字符一律
drop
```

IP directed-broadcast

默认情况下: 1. R3能PING 12.1.1.255
2. 当R2关掉ip dirrect-broadcast 后就PING 不通了


```
R1(config)#ip route 23.1.1.0 255.255.255.0 12.1.1.2
```

R3(config)#ip route 12.1.1.0 255.255.255.0 23.1.1.2
R2(config)#int e0/0.12
R2(config-if)#no ip directed-broadcast　　<====默认为关闭"directed-broadcast"
R2(config-if)#no ip unreachables　　<====使用此命令后,将不会再回"unreachables包"

active模式与passive模式

active (server send)

client ---------- (port:21) ----------> server
第一信道

client ---------- 应用层"port"命令 ----------> server
A . B . C . D . E . F
|<---IP 地址-->| Port: E*256+F

client <------- Source Port:20 Destination Port:E*256+F -------- server　　第二信道

passive (server receive)

client ---------- (port:21) ----------> server
第一信道

client -------- pasv (询问server是否支持Passive ?) --------> server

client <---------- A . B . C . D . E . F ---------- server
A . B . C . D . E . F
|<---IP 地址-->| Port: E*256+F

client -------- Source Port:random Destination Port:E*256+F --------> server　　第二信道
Client Port : random　　Server Port :random

Nagle算法
当在低速链路,要回TCP连接的ACK的大量包时,是非常消耗带宽的
R1(config)#service nagle　　<====使用此命令后,一个TCP连接上最多只能有一个未被确认的包

TCP intercept

1. Intercept

establish establish

client <----------> router <----------> server

2. watch

router listen in

client <----------> server

router监听到client与server的TCP连接超时,router即发送一个reset包,往server,重置连接.

SNRS

AAA

Security Network with Cisco Routers and Switches(SNRS)

AAA ----- authentication authorization accounting P305<<安全PIX防火墙>>

基本协议的拓扑图: Client--------NAS(network access system)------AAA

三大类的协议 : Client -------- NAS(network access system)

1. 登录(网管)路由器
2. 穿越路由器
3. 拨入路由器 (例：PPPOE 、EZVPN)

两大协议 : NAS ------ AAA

1. radius
2. tacacs+

两者之间的区别: P460 <<安全实验指南>>

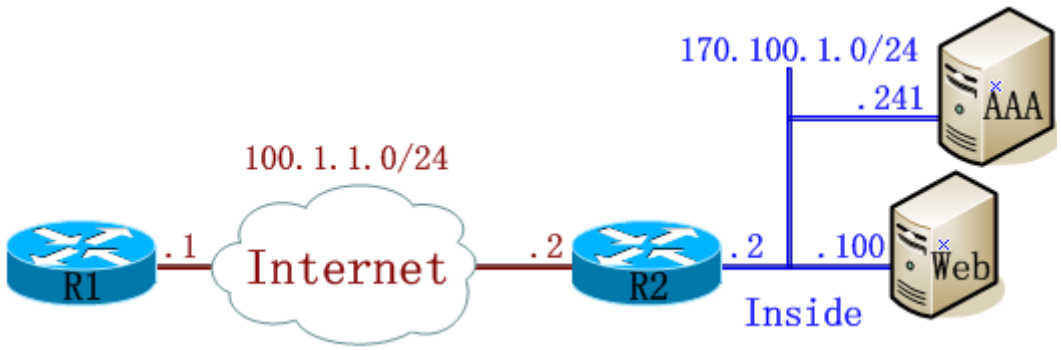
AV-pair: attribute value - pair

AAA服务器的安装: P308 <<安全PIX防火墙>>

远程访问AAA服务器: <http://170.100.1.241:2002/>
本地访问AAA服务器: <http://127.0.0.1:2002/>

配置:
1. 添加用户 : P313<<安全PIX防火墙>>
2. 添加组 : P325<<安全PIX防火墙>>
3. 配置radius服务器 P461<<安全实验指南>>

实验 1 :



R2允许telnet的四种方法:
1. R2(config-line)#no login <===不需要密码直接telnet
2. R2(config-line)#login local <====需要本地用户名和密码
3. R2(config-line)#login <====只需要密码
4. R2(config)#aaa new-model <====使用AAA服务器(一旦起用AAA,以上三条命令将配置不上)

好习惯:
R2(config)#aaa authentication login FOR_CON none <===P465<<安全实验指南>>
R2(config)#aaa authorization exec FOR_CON none
R2(config)#line console 0
R2(config-line)#login authentication FOR_CON
R1(config-line)#authorization exec FOR_CON

指定服务器:
R2(config)#tacacs-server host 170.100.1.241 key wolfcisco

测试:
R1#test aaa group tacacs+ test1 cisco new-code
用户名 密码 老版本不需要输

路由器的enable密码使用AAA认证:

```
R2(config)#aaa authentication enable default enable group tacacs+ none
<===慎用
```

要使用本地授权，先要使用本地认证:

```
R2(config)#user cisco privilege 15 password cisco
R2(config)#aaa authentication login VTY local
R2(config)#aaa authorization exec VTY local
R2(config)#line vty 0 4
R2(config-line)#login authentication VTY
R2(config-line)#authorization exec VTY
```

级别授权:

```
R2>show privilege                <===查看授权级别 P498<<安全实验指南>>
Current privilege level is 1
R2(config)#aaa authorization exec FOR_VTY group tacacs+
定义10级用户在特权模式使用的命令:
R2(config)#privilege exec level 10 configure terminal <===允许第10级的用户能用"configure terminal"
或者: privilege exec level 2 configure
      privilege exec level 2 configure terminal
定义10级用户在全局模式使用的命令:
R2(config)#privilege configure all level 2 router   <===允许第2级的用户能用"router"命令,及"router"命令的子命令,版本12.2T以上支持
R2(config)#privilege configure level 2 router     <===只允许第2级的用户能用"router"命令,不允许子命令
```

命令授权: 注意:应该从高级别授权开始

```
错误:
R2(config)#aaa authorization commands 0 FOR_VTY group tacacs+
R2(config)#line vty 0 4
R2(config-line)#authorization commands 0 FOR_VTY
对的:
R2(config)#aaa authorization commands 15 FOR_VTY group tacacs+
<====定义此用户的第15级命令去"tacacs+"服务器找授权
R2(config)#line vty 0 4
R2(config-line)#authorization commands 15 FOR_VTY
```

测试:

```
特权模式:
R2#reload                          <====="reload"为第15级命令
Command authorization failed.          <===授权失败:是因为"tacacs+"服务器没有对此命令进行授权
全局模式:
R2(config)#aaa authorization config-commands  <===没有此命令,全局模式下的
```

第15级命令不会去"tacacs+"服务器找授权
R2(config)#int e0/0
Command authorization failed.　　<===授权失败:是因为"tacacs+"服务器没有对此命令进行授权

审计:
R2(config)#aaa accounting exec default start-stop group tacacs+　　<===统计时间
R2(config)#aaa accounting commands 15 default start-stop group tacacs+　<===对15级做命令统计
R2(config)#aaa accounting commands 1 default start-stop group tacacs+

NAR:
　　NAR就是网络访问限制的意思
　　这个技术主要的目的是控制不同的 GROUP 或者 USER 能够通过什么 源IP 源端口 能够 登录到什么设备!
　　默认情况下任何GROUP或者USER可以通过任何 源IP 源端口 登录到任何设备

例:
环境: 我们创建一个USER: TEST
　　通过物理接口(fa1/0:202.100.1.10)
　　通过环回口(Lo0:1.1.1.1)
　　都可以成功访问

要求: 现在我们可以控制 TEST 这个USER只能登录到SERVER,并且源IP必须是202.100.1.0/24这个网络

　首先:我们需要在ACS上创建一个CLIENT

AAA Clients

AAA Client Hostname	AAA Client IP Address	Authenticate Using
SERVER	170.1.1.1	TACACS+ (Cisco IOS)

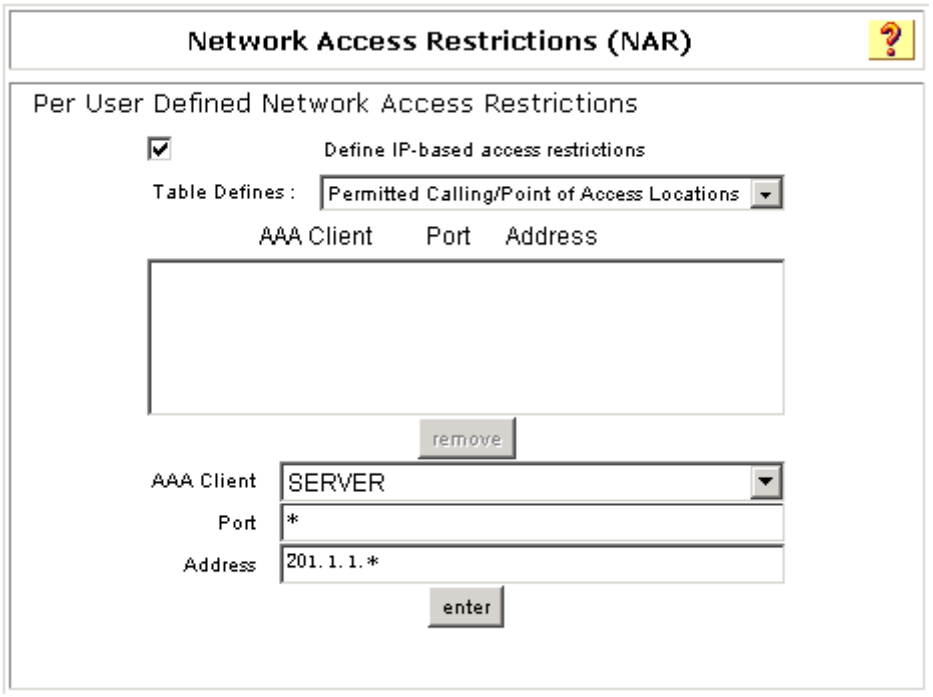
　然后:在 Interface configuration ---- Advanced Options ---- User ---- Level Network
 Access Restrictions

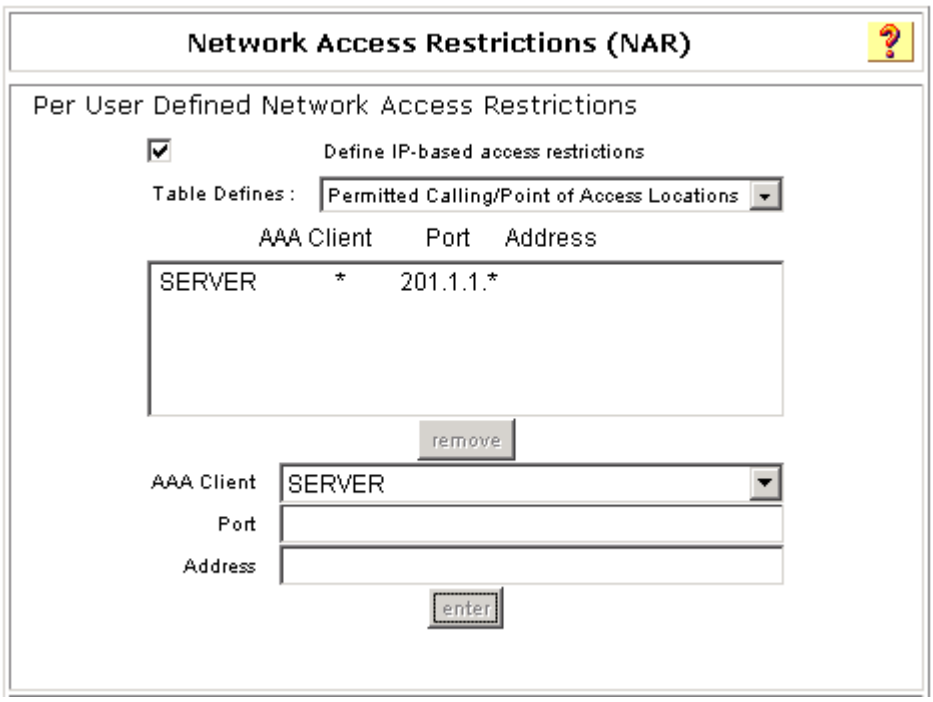
Advanced Options

Note: Only the selected options will appear in the user interface.

- [ ] Per-user TACACS+/RADIUS Attributes
- [ ] User-Level Shared Network Access Restrictions
- [x] User-Level Network Access Restrictions

　最后: 到USER: TEST 下配置





AAA Client ： 控制这个用户能够登录的设备
Port ： 控制这个用户登录时候使用的源端口
Address ： 控制这个用户登录时候所使用的源IP

CBAC

IOS防火墙的三大组件:　　IOS防火墙的理念及特性:P619 <<安全实验指南>>

1.CBAC ----- Context-Based Access Control　　<===CBAC是工作在List之后的

2.Authentication Proxy　　<=====Authentication Proxy是工作在List之前

3.IPS ------ Intrusion Prevention System

CBAC的概述: P624 <<安全实验指南>>

实验 1:　环境inside和outside路由器可相互telnet

配置:

初始命令:

```
gateway(config)#ip inspect audit-trail      <===打开审计(默认关闭)
gateway(config)#ip inspect alert-off         <====关闭告警(默认打开)
gateway(config)#logging on                <====打开LOG日志(默认打开)
gateway(config)#logging 61.1.14.100          <====指定syslog服务器
R2_gateway#show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0
flushes, 0 overruns, xml disabled)
    Console logging: level debugging, 14 messages logged, xml disabled
```

```
    Monitor logging: level debugging, 0 messages logged, xml disabled
    Buffer logging: disabled, xml disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 19 message lines logged
        Logging to 61.1.14.88, 0 message lines logged, xml disabled
        Logging to 61.1.14.100, 0 message lines logged, xml disabled

Define inspection rules for application protocols:
R2_gateway(config)#ip inspect name CBAC ?        <===以下就是可以识别监控
的protocols
  cuseeme      CUSeeMe Protocol
  fragment     IP fragment inspection
  ftp          File Transfer Protocol
  h323         H.323 Protocol (e.g, MS NetMeeting, Intel Video Phone)
  http         HTTP Protocol
  icmp         ICMP Protocol
  netshow      Microsoft NetShow Protocol
  rcmd         R commands (r-exec, r-login, r-sh)
  realaudio    Real Audio Protocol
  rpc          Remote Prodedure Call Protocol
  rtsp         Real Time Streaming Protocol
  sip          SIP Protocol
  skinny       Skinny Client Control Protocol
  smtp         Simple Mail Transfer Protocol
  sqlnet       SQL Net Protocol
  streamworks  StreamWorks Protocol
  tcp          Transmission Control Protocol
  tftp         TFTP Protocol
  udp          User Datagram Protocol
  vdolive      VDOLive Protocol
R2_gateway(config)#ip inspect name CBAC http ?
  alert        Turn on/off alert
  audit-trail  Turn on/off audit trail
  java-list    Specify a standard access-list to apply the Java blocking.
If
               specified, MUST appear directly after option "http"
  timeout      Specify the inactivity timeout time
  urlfilter    Specify URL filtering for HTTP traffic
gateway(config)#ip inspect name CBAC tcp
gateway(config)#ip inspect name CBAC udp
gateway(config)#ip inspect name CBAC http

Define outside flow do not access inside :
```

```
gateway(config)#ip access-list extended ACLIN
gateway(config-ext-nacl)#deny ip any any
gateway(config)#int e0/0.202
gateway(config-subif)#ip access-group ACLIN in
gateway(config-subif)#ip inspect CBAC out          <===application ruless
of interface

fragment防御:
gateway(config)#ip inspect name CBAC fragment

gateway#show ip access-lists       <===12.4以上的版本,使用此命令看不到
Extended IP access list ACLIN
    permit tcp host 202.100.24.2 eq telnet host 61.1.14.1 eq 55798 (8
matches)
    10 deny ip any any (15 matches)

gateway#show ip inspect sessions
Established Sessions
 Session 62BF013C (61.1.14.1:55798)=>(202.100.24.2:23) tcp SIS_OPEN

------------------------------------------------------------------------
gateway#show ip inspect all
Session audit trail is enabled
Session alert is dlsabled
以下三行是对DOS攻击的一些防御:
一分钟之内的半开连接数:
    当一分钟之内半开连接数达到500,就开始删除一些半开连接.当半开连接数减低
至400时,就停止删除.
命令:
    gateway(config)#ip inspect one-minute high 800
    gateway(config)#ip inspect one-minute low 600
one-minute (sampling period) thresholds are [400:500] connections
总的半开连接数:
    当总的半开连接数达到500,就开始删除一些半开连接.当半开连接数减低至400
时,就停止删除.
命令:
    gateway(config)#ip inspect max-incomplete high 1000
    gateway(config)#ip inspect max-incomplete low 500
max-incomplete sessions thresholds are [400:500]
基于主机的TCP协议的半开连接数:
当"Block-time"为"零"时:  当同一目标IP地址超过了50个半开连接,就会删除老的
连接.
当"Block-time"为"1"时:   当同一目标IP地址超过了50个半开连接,就会删除基于
同一目标IP地址的所有半开连接数,并且在一分钟之内,不允许所有主机的TCP连接.
```

```
    gateway(config)#ip inspect tcp max-incomplete host 100 block-time 1
    gateway(config)#ip inspect tcp max-incomplete host 100 block-time 0
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
```

命令:

```
  gateway(config)#ip inspect ?
  alert-off       Disable alert
  audit-trail     Enable the logging of session information (addresses
and
                  bytes)
  dns-timeout     Specify timeout for DNS
  hashtable-size  Specify size of hashtable
  max-incomplete  Specify maximum number of incomplete connections before
                  clamping
  name            Specify an inspection rule
  one-minute      Specify one-minute-sample watermarks for clamping
  tcp             Config timeout values for tcp connections
  udp             Config timeout values for udp flows
Inspection Rule Configuration
 Inspection name CBAC
    tcp alert is off audit-trail is on timeout 3600
    udp alert is off audit-trail is on timeout 30
    http alert is off audit-trail is on timeout 3600
Interface Configuration
 Interface Ethernet0/0.202
  Inbound inspection rule is not set
  Outgoing inspection rule is CBAC
    tcp alert is off audit-trail is on timeout 3600
    udp alert is off audit-trail is on timeout 30
    http alert is off audit-trail is on timeout 3600
  Inbound access list is ACLIN
  Outgoing access list is not set
Established Sessions
 Session 62BF013C (61.1.14.1:55798)=>(202.100.24.2:23) tcp SIS_OPEN
```

PAM ---- Port-to-Application Mapping

```
gateway#show ip port-map          <===查看默认端口对应相应的服务
Default mapping: dns              port 53                 system defined
Default mapping: vdolive          port 7000               system defined
Default mapping: sunrpc           port 111                system defined
Default mapping: netshow          port 1755               system defined
Default mapping: cuseeme          port 7648               system defined
```

```
Default mapping: tftp              port 69                  system defined
Default mapping: https             port 443                 system defined
Default mapping: rtsp              port 8554                system defined
Default mapping: realmedia         port 7070                system defined
Default mapping: streamworks       port 1558                system defined
Default mapping: ftp               port 21                  system defined
Default mapping: telnet            port 23                  system defined
Default mapping: rtsp              port 554                 system defined
Default mapping: h323              port 1720                system defined
Default mapping: sip               port 5060                system defined
Default mapping: smtp              port 25                  system defined
Default mapping: http              port 80                  system defined
Default mapping: msrpc             port 135                 system defined
Default mapping: exec              port 512                 system defined
Default mapping: login             port 513                 system defined
Default mapping: sql-net           port 1521                system defined
Default mapping: shell             port 514                 system defined
Default mapping: skinny            port 2000                system defined
Default mapping: mgcp              port 2427                system defined

gateway(config)#ip port-map ftp port 2121     <===为FTP增加新的端口号
gateway(config)#ip port-map http port 8080

gateway(config)#ip port-map ftp port 80 list 50     <===基于某台主机定义已有标准的端口号
gateway(config)#access-list 50 permit 61.1.14.254

gateway#show ip port-map ftp
Default mapping: ftp               port 2121                user defined
Default mapping: ftp               port 21                  system defined
Host specific:   ftp               port 80      in list 80  user defined
```

基于List允许某些主机不做Java Blocking ：

功能：只是将页面的JAVA功能过滤掉，不是删除页面

```
gateway(config)#ip inspect name CBAC http java-list 1
gateway(config)#access-list 1 permit 61.1.1.0 0.0.0.255
```

将网关发过来的URL送去URL服务器上去做过滤 ：

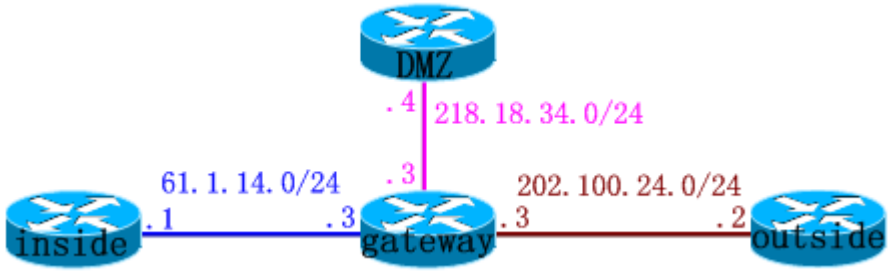
功能：禁止查看不被允许的网页

```
gateway(config)#ip inspect name CBAC http java-list 1 urlfilter
gateway(config)#ip urlfilter server vendor websense 61.1.14.68     <===定义过滤URL的服务器

gateway(config)#ip inspect name CBAC icmp     <===允许PING包通过(新版本)；
```

老版本需要写list

gateway(config)#no ip inspect　　<===将所有的CBAC配置清空/重置所有的时间及门限值/删除所有的会话/删除所有的动态列表

实验 2:



不允许DMZ服务器发起连接：

```
R3_gateway(config)#ip access-list extended DMZ
R3_gateway(config-ext-nacl)#deny ip any any
R3_gateway(config)#int e0/0.218
R3_gateway(config-subif)#ip access-group DMZ in
```

允许Inside的主机往任何方向发起telnet连接：

```
R3_gateway(config)#ip inspect name CBAC_R1 tcp
R3_gateway(config)#int e0/0.61
R3_gateway(config-subif)#ip inspect CBAC_R1 in
```

允许Outside的主机往DMZ区域发起telnet连接，但不允许往Inside区域发起连接：

```
R3_gateway(config)#ip inspect name CBAC_R2 tcp
R3_gateway(config)#int e0/0.218
R3_gateway(config-subif)#ip inspect CBAC_R2 out
R3_gateway(config)#ip access-list extended OUT_ACL
R3_gateway(config-ext-nacl)#permit tcp any host 218.18.34.4 eq 23
R3_gateway(config-subif)#ip access-group OUT_ACL in
```

Authentication Proxy

Authentication Proxy支持的协议:

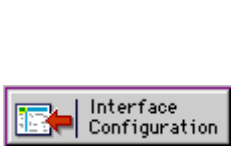
1. HTTP
2. HTTPS　　<===(12.3T或12.4的版本才支持)
3. FTP
4. TELNET

通过TACACS+/RADIUS协议动态地对穿越路由器的用户进行认证和授权,可以工作在InBound或OutBound方向.

因为Authentication Proxy是工作在List之前,所以不需要在List上放行所要认证/授权的流量,但一定要放行AAA服务器的流量.

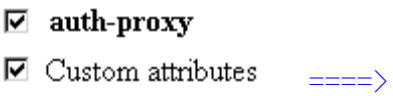
Greate Authentication Proxy Server :

1 .



====>

2 .授权:



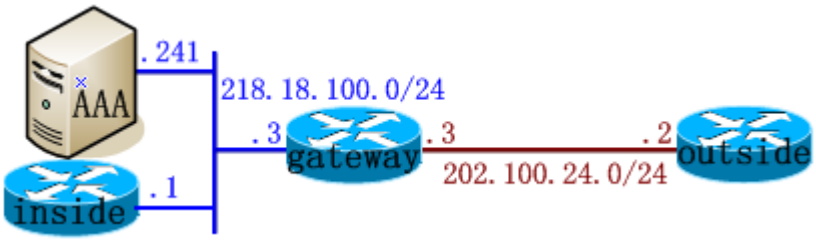
====>

```
proxyacl#1=permit tcp any any
priv-lvl=15
```

红色字为格式:　priv-lvl=15

proxyacl#n=permit tcp any any　　<====源必须是"any",当认证成功后会将"any",换成认证主机的IP地址.目标是根据需求.

实验 1 ： 默认环境:R1能TELNET到R2



步骤:

1. 首先到AAA服务器CLI界面上写: `>route add 202.100.24.2 218.18.100.3`
2. 将R2设置成HTTP服务器

```
R2_out(config)#ip http server
```

3. R2起用本地认证

```
R2_out(config)#ip http authentication local
R2_out(config)#username cisco privilege 15 password cisco
```

4. 起用AAA后:

```
R3_gateway(config)#aaa authentication login default group tacacs+
R3_gateway(config)#aaa authorization auth-proxy default group tacacs+
R3_gateway(config-ext-nacl)#permit tcp host 218.18.100.241 eq 49 host
218.18.100.3
R3_gateway(config-ext-nacl)#deny ip any any
R3_gateway(config-ext-nacl)#int e0/0.170
R3_gateway(config-subif)#ip access-group ACLIN in
R3_gateway(config)#ip http server
R3_gateway(config)#ip http authentication aaa
R3_gateway(config)#access-list 1 deny any      <===定义不允许通过HTTP
网管网关
R3_gateway(config)#ip http access-class 1       <===调用
R3_gateway(config)#ip auth-proxy auth-proxy-banner http &
Enter TEXT message.  End with the character '&'.
welcome HTTP &
R3_gateway(config)#ip auth-proxy name AUTH http
R3_gateway(config)#int e0/0.170
R3_gateway(config-subif)#ip auth-proxy AUTH

R3_gateway#clear ip auth-proxy cache *
R3_gateway#show ip auth-proxy
Authentication Proxy Cache
 Client IP 218.18.100.241 Port 4160, timeout 60, state ESTAB
R3_gateway#show access-lists
Standard IP access list 1
    10 deny   any
Extended IP access list AUTH
     permit tcp host 218.18.100.241 host 202.100.24.2 (14 matches)
    10 permit tcp host 218.18.100.241 eq tacacs host 218.18.100.3 (118
```

```
matches)
    20 permit icmp any any
    30 deny ip any any (180 matches)
```

IOS IPS

配置:
IOS_IPS(config)#ip ips sdf location disk2:attack-drop.sdf <==指明sdf文件(包含signature)的存放位置
IOS_IPS(config)#ip ips sdf builtin <====使用路由器内建的signature
IOS_IPS(config)#ip ips name IPSRULE <===建立一个Rule,"IPSRULE"是"Rule"的名字
IOS_IPS(config)#ip ips signature 1000 disable <===针对某些signature,将它禁用
IOS_IPS(config-if)#ip ips IPSRULE in <===将Rule应用于某个接口

Security Device Event Exchange(SDEE) Protocol ---- 类似事件查看器
IOS_IPS(config)#ip ips notify SDEE <===将告警信息加密传送
IOS_IPS(config)#ip ips notify log <===将告警信息发送到syslog服务器上

Instructs the router to drop all packets until the signature engine is built and ready to scan traffic :
IOS_IPS(config)#ip ips fail closed <====默认配置(即没有初始化完成前,所有包将扔掉)

IOS_IPS#show ip ips configuration
Configured SDF Locations: none
Builtin signatures are enabled and loaded
Last successful SDF load time: 05:40:43 UTC Mar 2 2002
IPS fail closed is disabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is disabled
Total Active Signatures: 132
Total Inactive Signatures: 0

```
Signature 1000:0 disable
Signature 1107:0 disable
IPS Rule Configuration
 IPS name IPS
Interface Configuration
 Interface Ethernet0/0
  Inbound IPS rule is IPS
  Outgoing IPS rule is not set
-----------------------------------------------
IOS_IPS#show ip ips signatures        <====显示所有的signatures
Builtin signatures are configured
Builtin signatures are loaded
Cisco SDF release version S46.0
Trend SDF release version V0.0
*=Marked for Deletion  Action=(A)larm,(D)rop,(R)eset   Trait=AlarmTraits
MH=MinHits             AI=AlarmInterval               CT=ChokeThreshold
TI=ThrottleInterval    AT=AlarmThrottle               FA=FlipAddr
WF=WantFrag
Signature Micro-Engine: OTHER (3 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version
 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- --
-------
  1202:0      Y   A    HIGH     0      0     0   100    15 FA  N  Y S37
  1206:0      Y   A    INFO     0      0     0   100    15 FA  N  Y S37
  3050:0      Y   A    HIGH     0      0     0     0    15 FA  N    S37
Signature Micro-Engine: STRING.UDP (1 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version
 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- --
-------
  4100:0      Y   A    HIGH     0      0     0     0    15 FA  N    S37
Signature Micro-Engine: STRING.TCP (3 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version
 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- --
-------
  3150:0      Y   A    INFO     0      1     0     0    15 FA  N    S37
  3151:0      Y   A    INFO     0      1     0     0    15 FA  N    S37
  3152:0      Y   A     MED     0      1     0     0    15 FA  N    S37
Signature Micro-Engine: SERVICE.FTP (2 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF
Version
 ----------- -- ------ ---- -----  ----- ----- ----- ----- -- -- --
```

```
--------
  3153:0       Y    A      MED      0        0      0      0     15 FA  N     S37
  3154:0       Y    A      MED      0        0      0      0     15 FA  N     S37
......

------------------------------------------------------------------------------
-------------
```

IOS IDS

IOS IDS的配置:

```
IOS_IDS(config)#ip audit smtp spam 50
IOS_IDS(config)#ip audit notify log
IOS_IDS(config)#ip audit notify nr-director
IOS_IDS(config)#ip audit po local hostid 100 orgid 10100
IOS_IDS(config)#ip audit po remote hostid 200 orgid 10100 rmtaddress
100.1.1.100 localaddress 12.1.1.2
IOS_IDS(config)#ip audit name IOSIDS info action alarm
IOS_IDS(config)#ip audit name IOSIDS attack action alarm drop reset
IOS_IDS(config)#int e0/0
IOS_IDS(config-if)#ip audit IOSIDS in
IOS_IDS(config)#ip audit name IOSIDS info list 1 action alarm drop
IOS_IDS(config)#access-list 1 permit host 12.1.1.1
```

```
R1#ping 12.1.1.2 size 1025

IOS_IDS#show ip audit all
Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm
Default threshold of recipients for spam signature is 50
PostOffice:HostID:100 OrgID:10100 Msg dropped:0
          :Curr Event Buf Size:100  Configured:100
Host ID:200, Organization ID:10100, SYN pkts sent:95,
ACK pkts sent:0, Heartbeat pkts sent:0, Heartbeat ACK pkts sent:0,
Duplicate ACK pkts received:0, Retransmission:0, Queued pkts:0
  ID:1 Dest:100.1.1.100:45000 Loc:12.1.1.2:45000 T:5 S:SYN SENT
Audit Rule Configuration
 Audit name IOSIDS
    info acl list 1 actions alarm drop
    attack actions alarm drop reset
Interface Configuration
 Interface Ethernet0/0
  Inbound IDS audit rule is IOSIDS
    info acl list 1 actions alarm drop
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
```

TCP Intercept

TCP Intercept

Layer 2 security

CAM table overflow

耗尽CAM表中的"MAC地址对应端口"的条目,使交换机变成"HUB"

解决:端口安全

Media Access Control(MAC) Address Spoofing

欺骗交换机

解决:端口安全

ARP Spoofing

欺骗PC机

解决:可使用"ARP -A"去查询"IP"与"MAC地址"的绑定,或使用"ARP -S"

DHCP ---- Dynamic Host Configuration Protocol

它可以为局域网内的主机自动分配IP地址、子网掩码、网关、WINS、DNS等网络通信所必须的参数。DHCP是基于C/S模式工作的，客户端通过向服务器端提出请求并从服务器端接收相应的配置信息，从而完成配置工作。

要成功获得网络中的配置信息，需要成功处理4个报文，如果某一个报文出错，客户端均不能成功从服务器处获得配置信息。这4个报文是：

1.DHCP Discover：Discover报文是客户端发出的第一个报文，该报文被发往广播地址255.255.255.255，客户端IP地址使用0.0.0.0，并在报文中包含一个特定的事务ID；

2.DHCP Offer：Offer报文是服务器端收到Discover报文后对客户端的响应，该报文中包含了客户端所请求的相应配置参数（IP、掩码、DNS、WINS等）。

3.DHCP Request：客户端收到Offer报文，并检查所得到的配置参数，如果与请求的一致，则发送一个Request报文，请求使用这些配置信息。

4.DHCP Ack：服务器端在收到Request报文后，向客户端返回一个Ack的报文，通知客户端的请求都已完成，客户端可以立即开始使用相关的配置参数。

DHCP通过数据链路层的广播模式工作，在整个DHCP的4个报文处理过程中，数据包全部发往广播地址255.255.255.255。在四个报文中，均包含有Discover报文中所指定的事务ID，而不同的DHCP客户端主机则通过该ID值判断网络中发送的DHCP报文是否是发往自己的。

DHCP "starvation"

```
Switch(config)#ip dhcp snooping                    <===起用"DHCP Snooping"
Switch(config)#ip dhcp snooping vlan 100      <====在VLAN 100 中起用"DHCP Snooping"
Switch(config)#int f0/10      <===此接口为连接DHCP服务器的接口
Switch(config)#switch access vlan 100       <====如果DHCP Server和DHCP Client不在同一VLAN,则两个VLAN都需要起DHCP SNOOPING
Switch(config-if)#ip dhcp snooping trust       <====使用此命令,服务器才可以下发IP地址
Switch(config-if)#ip dhcp snooping limit rate 3000  <===限速(限制发送速率为:3000包/秒)
```

Dynamic ARP Inspection

```
Switch(config)#ip arp inspection vlan 1,3-5,9-11
Switch(config)#ip arp inspection validate ?
```

```
  dst-mac  Validate destination MAC address
  ip       Validate IP addresses
  src-mac  Validate source MAC address
Switch(config-if)#ip arp inspection trust
```

Identity Based Network Services

基于身份的认证:可以基于用户名/密码,可以基于802.1X,可以基于病毒库.....

802.1X ---- 二层的认证技术

802.1X的认证方式: 用户名/密码，MAC地址；支持授权；支持DHCP

EAP ---- Extensible Authentication Protocol

Extension of PPP to provide additional authentication features

EAP-MD5:

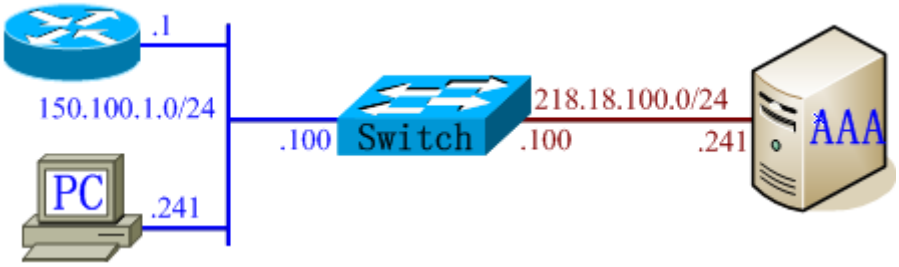
EAP-TLS(based on X.509 certificates): Requires server and client certificates

TLS---Transport Layer Security

LEAP(EAP-Cisco Wireless)---Lightweight Extensible Authentication Protocol: 专用于无线

PEAP(Protected EAP): Requires server certificates only ,用于无线

实验 1：



配2950的SVI：

```
Switch(config)#int vlan 1
Switch(config-if)#ip add 218.18.100.100 255.255.255.0
Switch(config-if)#no sh
```

认证:

在Switch中起用AAA服务器：使用radius(IETF)

```
Switch(config)#aaa authentication dot1x default group radius    <===使用了default,一定要放行CONSOLE和VTY和AUX,但换机上没有AUX口
Switch(config)#dot1x system-auth-control      <===打开802.1X
Switch(config)#int fastEthernet 0/18        <===对这个接口做认证
Switch(config-if)#switchport mode access
Switch(config-if)#dot1x port-control auto
Switch(config-if)#dot1x ?
  auth-fail          Configure Authentication Fail values for this port
  control-direction  set the control-direction value
```

```
  default            Configure Dot1x with default values for this port
  guest-vlan         Configure Guest-vlan on this interface  <===当认证不
成功,将会划进"guest-vlan"
  host-mode          Set the Host mode for 802.1x on this interface  <===
接入一台或多台PC
  max-reauth-req     Max No. of Retries to supplicant
  max-req            Max No. of Retries to Radius
  port-control       set the port-control value
  reauthentication   Enable or Disable Reauthentication for this port
<===打开重认证
  timeout            Various Timeouts
```

手工重认证:

必需要在接口下已经开启"Switch(config-if)#dot1x reauthentication"

```
Switch#dot1x re-authenticate interface fastEthernet 0/19
```

在PC机上配置:

EAP类型:MD5质询

win2000可能需要:

打开"Control Panel" ===>"Administrative Tools"===>"Services" ===>"Wireless Configuration Properties"===>"General" ===>"start"

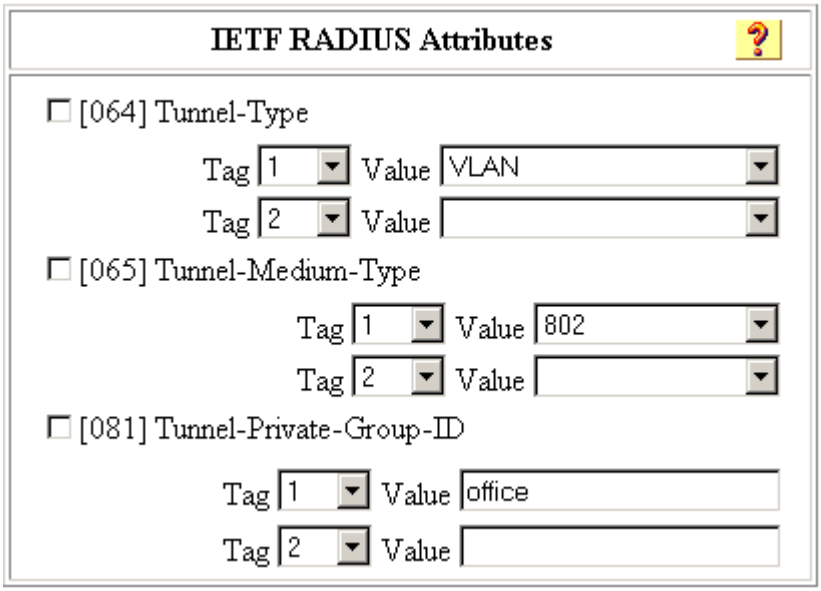
```
Switch#show dot1x all
Sysauthcontrol              Enabled
Dot1x Protocol Version            2
Critical Recovery Delay         100
Critical EAPOL              Disabled

Dot1x Info for FastEthernet0/16
----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0
```

授权:

```
Switch(config)#aaa authorization network default group radius
```

创建"VLAN ID"和"VLAN 名字"

基于用户名和密码动态划分VLAN,在AAA服务器上一定要勾选以下三项:



[064]和[065]的填写内容为固定的;[081]的填写内容:"office"是vlan的名字(cisco推荐),但也可以写vlan号

```
Switch(config-if)#switchport trunk allowed vlan ?
```

<====限制某些VLAN通过trunk

```
  WORD    VLAN IDs of the allowed VLANs when this port is in trunking
mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list
```

建议:关闭边界接口的CDP功能.

SDM

SDM -------- Security Device Manager
使用了两个协议:TCP443端口和SSH

设置时间: 时间不对,对VPN会有影响
```
Router(config)#clock timezone GMT +8
Router#clock set 18:56:00 9 march 2007
Router(config)#ip http secure-server          <===一旦输入此命令,"show run"时,会出现自签名证书
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

打开HTTP的本地认证:
```
Router(config)#username wolf privilege 15 password wolf
Router(config)#ip http authentication local
Router(config)#line vty 0 4
Router(config-line)#login local            <==="SSH"必须是"login local"
Router(config-line)#transport input ssh telnet
```

other

基本安全配置详解

Cisco IOS基本安全配置详解

no ip domain-lookup 关闭域名查询
no cdp run 禁用cdp
no ip http server 禁用http server，这玩意儿的安全漏洞很多的
no ip source-route 禁用IP源路由，防止路由欺骗
no service finger 禁用finger服务
no ip bootp server 禁用bootpe服务
no service udp-small-s 禁用一堆小的udp服务
no service tcp-small-s 禁用一堆小的tcp服务

service timestamp log datetime localtime 配置时间戳为datetime方式，使用本地时间
logging 192.168.0.1 向192.168.0.1发送log
logging 192.168.0.2 向192.168.0.2发送log
snmp-server community HSDxdf ro 98 配置snmp只读通讯字，并只允许access-list 98的主机进行通讯
service password-encryption 启用加密服务，将对password密码进行加密
enable secret asdfajkls 配置强加密的特权密码

no enable password 禁用弱加密的特权密码
no access-list 99 在配置一个新的acl前先清空该ACL

access-list 1 deny any log log参数说明在有符合该条件的条目时产生一条logo信息
no access-list 1 在配置一个新的acl前先清空该ACL
access-list 1 deny any log log参数说明在有符合该条件的条目时产生一条logo信息

line vty 0 4
access-class 1 in 使用acl 1来控制telnet的源地址
login
password 0 rererkerlf 配置telnet密码
exec-timeout 2 0 配置虚终羰超时参数，这里是2分钟
!
line con 0
login
password 0 aeroer 配置console口的密码
exec-timeout 2 0 配置console口超时参数，这里是两分钟
!
line aux 0
transport input none
password 0 asfdkalsfj
no exec
exit
banner motd # 配置提示信息
This is a private system operated for UltraTeam.
Authorization from UltraTeam is required to use this system
Use by unauthorized persons is prohibited
#
!
clock timezone PST-8 设置时区
ntp authenticate 启用NTP认证
ntp authentication-key 1 md5 uadsf 设置NTP认证用的密码，使用MD5加密。需要和ntp server一致
ntp trusted-key 1 可以信任的Key.
ntp acess-group peer 98 设置ntp服务，只允许对端为符合access-list 98条件的主机
ntp server 192.168.0.1 key 1 配置ntp server，server为192.168.0.1，使用1号key做为密

码
!
interface vlan 10
ip address 10.1.10.254 255.255.255.0
ip ospf authentication message-digest 启用ospf邻接过程中的认证
ip ospf message-digest-key md5 7 24d10564152140a7e 配置ospf认证key，要求所有邻居相同
ip ospf priority 3 提高ospf的优先级，便于成为DR

CISCO IOS中的快捷键

CISCO IOS中的快捷键

<ESC><B>:光标向左移至一个单词的首字符。操作：先按下ESC键，松开；然后再按下B键
<ESC><F>:光标向右移至一个单词的末尾。操作：同上
<Ctrl＋A>：将光标移至命令行开头。操作：ctrl和A键同时按下
<Ctrl＋E>：将光标移至命令行末尾。操作：同上
<Ctrl＋W>或<ESC><DELETE>:将光标从左侧字符开始向左一直删除到一个单词的开头
<ESC><D>:从光标右侧一直删除到一个单词结尾
<Ctrl＋U>或<Ctrl＋X>:将光标从左侧字符开始一直删除到命令行开头
<Ctrl＋U>：可删除一整行
<Ctrl＋K>：从光标字符开始一直删除到命令行末尾
<Ctrl＋Y>：将删除缓冲区里的最后一个条目粘贴回命令行
<Ctrl＋R>或<Ctrl＋L>：重现当前命令行
<ESC><U>:从光标位置到词尾所有字符大写
<ESC><L>:从光标位置到词尾所有字符小写
<ESC><C>:将光标位置的字符变成大写，随后光标移至词尾

```
Gz_Sec_Rack02>show sessions
Conn Host                Address             Byte  Idle Conn Name
   1 pix1                1.1.1.1                0     0 pix1
*  2 sw1                 1.1.1.1                0     0 sw1

Gz_Sec_Rack02>disconnect 1
Closing connection to pix1 [confirm]
```

Question

question

1. guest vlan 做不成功?

2. pki 的远程VPN做不成功

3. 处理arp病毒，其主要步骤有两步:
(1) 运行 tracert -d www.163.com 找出作祟的主机IP地址。
(2) 设置与作祟主机相同的IP，然后造成IP地址冲突，使中毒主机报警然后找到这个主机。