电缆上的 IPSec 示例配置和调试

目录

简介

Internet协议安全性(IPsec)是保证在IP网络的安全专用通信开放标准的框架。基于互联网工程任务组(IETF)开发的标准， IPsec保证机密性、数据通信完整性和真实性在公共IP网络间的。IPsec为基于标准的提供一个必要的组件，灵活的解决方案实施全网安全策略。

本文提供IPsec配置示例在两Cisco电缆调制解调器之间的。此配置创建在间一个有线网络的一个加密隧道在两个Cisco UBR9XX系列cable modem路由器之间。两网络之间的所有流量加密。但是为其他网络注定的流量允许通过未加密。对于小型办公室、家庭办公室用户，这允许虚拟专用网络的创建在间有线网络的。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

调制解调器必须依照这些需求配置在两电缆调制解调器的IPsec ：

- Cisco UBR904、uBR905或者uBR924在路由模式
- IPsec 56特性组
- Cisco IOS软件版本12.0(5)T或以上

另外，您必须有有线调制解调器终端系统(CMTS)，是所有有线电缆数据服务接口规范(DOCSIS) -兼容头端有线路由器，例如Cisco UBR7246、Cisco UBR7223或者思科uBR7246VXR。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

背景理论

在本文的示例使用一个uBR904有线调制解调器、一个uBR924有线调制解调器和一uBR7246VXR CMTS。电缆调制解调器运行Cisco IOS软件版本12.1(6)，并且CMTS运行Cisco IOS软件版本12.1(4)ec。

注意： 此示例完成与在电缆调制解调器的手动配置到控制台端口。如果自动化进程通过DOCSIS配置文件进行(ios.cfg脚本创建与IPSec配置)然后访问列表100和101不可能使用。这是因为简单网络管理协议(SNMP) docsDevNmAccess表的Cisco实施使用Cisco IOS访问列表。它建立每个接口一访问列表。在uBR904，前两访问列表通常使用924和905 (100和101)。在支持通用串形总线(USB)，类似CVA120，三访问列表的有线调制解调器上使用(100， 101和102)。

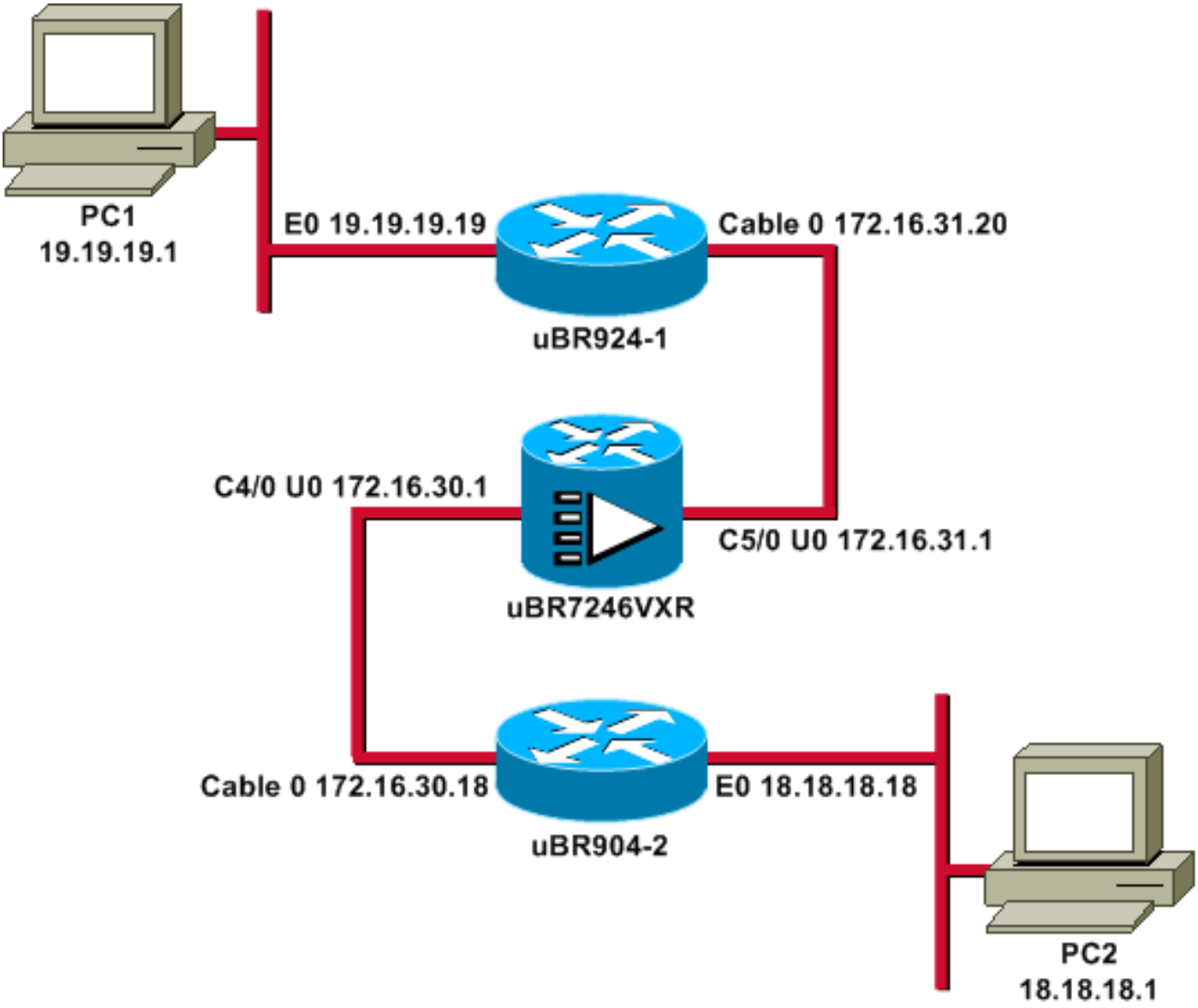
配置

本部分提供有关如何配置本文档所述功能的信息。

注意： 请使用命令查找工具(仅限注册用户)找到关于in命令的其他信息本文。

网络图

本文档使用以下网络设置：



注意： 所有IP地址在此图表中有一24位掩码。

配置

本文档使用以下配置：

- uBR924-1
- uBR904-2
- uBR7246VXR

uBR924-1

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ubr924-1
!
enable password ww
!
!
!
!
```

```
clock timezone - -8
ip subnet-zero
no ip finger
!
ip audit notify log
ip audit po max-events 100
!
!
crypto isakmp policy 10 !--- Creates an Internet Key
Exchange (IKE) policy with the specified priority !---
number of 10. The range for the priority is 1 to 10000,
where 1 is the !--- highest priority. This command also
enters Internet Security Association !--- and Key
Management Protocol (ISAKMP) policy configuration
command mode. hash md5 !--- Specifies the MD5 (HMAC
variant) hash algorithm for packet authentication.
authentication pre-share !--- Specifies that the
authentication keys are pre-shared, as opposed to !---
dynamically negotiated using Rivest, Shamir, and Adelman
(RSA) public !--- key signatures. group 2 !--- Diffie-
Hellman group for key negotiation. lifetime 3600 !---
Defines how long, in seconds, each security association
should exist before !--- it expires. Its range is 60 to
86400, and in this case, it is 1 hour. crypto isakmp key
mykey address 18.18.18.18 !--- Specifies the pre-shared
key that should be used with the peer at the !---
specific IP address. The key can be any arbitrary
alphanumeric key up to !--- 128 characters. The key is
case-sensitive and must be entered identically !--- on
both routers. In this case, the key is mykey and the
peer is the !--- Ethernet address of uBR904-2 . ! crypto
IPSec transform-set TUNNELSET ah-md5-hmac esp-des !---
Establishes the transform set to use for IPsec
encryption. As many as !--- three transformations can be
specified for a set. Authentication Header !--- and ESP
are in use. Another common transform set used in
industry is !--- esp-des esp-md5-hmac. ! crypto map
MYMAP local-address Ethernet0 !--- Creates the MYMAP
crypto map and applies it to the Ethernet0 interface.
crypto map MYMAP 10 ipsec-isakmp !--- Creates a crypto
map numbered 10 and enters crypto map configuration
mode. set peer 18.18.18.18 !--- Identifies the IP
address for the destination peer router. In this case,
!--- the Ethernet interface of the remote cable modem
(ubr904-2) is used. set transform-set TUNNELSET !---
Sets the crypto map to use the transform set previously
created. match address 101 !--- Sets the crypto map to
use the access list that specifies the type of !---
traffic to be encrypted. !--- Do not use access lists
100, 101, and 102 if the IPsec config is !--- downloaded
through the ios.cfg in the DOCSIS configuration file. !
! ! ! voice-port 0 input gain -2 output attenuation 0 !
voice-port 1 input gain -2 output attenuation 0 ! ! !
interface Ethernet0 ip address 19.19.19.19 255.255.255.0
ip rip send version 2 ip rip receive version 2 no ip
route-cache no ip mroute-cache ! interface cable-modem0
ip rip send version 2 ip rip receive version 2 no ip
route-cache no ip mroute-cache cable-modem downstream
saved channel 525000000 39 1 cable-modem mac-timer t2
40000 no cable-modem compliant bridge crypto map MYMAP
!--- Applies the previously created crypto map to the
cable interface. ! router rip version 2 network 19.0.0.0
network 172.16.0.0 ! ip default-gateway 172.16.31.1 ip
classless ip http server ! access-list 101 permit ip
```

```
19.19.19.0 0.0.0.255 18.18.18.0 0.0.0.255 !--- Access
list that identifies the traffic to be encrypted. In
this case, !--- it is setting traffic from the local
Ethernet network to the remote !--- Ethernet network.
snmp-server manager ! line con 0 transport input none
line vty 0 4 password ww login ! end
```

另一个有线调制解调器的配置是非常类似的，大多注释在先前配置里如此省略。

uBR904-2

```
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ubr904-2
!
enable password ww
!
!
!
!
!
clock timezone - -8
ip subnet-zero
no ip finger
!
!
!
crypto isakmp policy 10 hash md5 authentication pre-
share group 2 lifetime 3600 crypto isakmp key mykey
address 19.19.19.19 ! ! crypto IPSec transform-set
TUNNELSET ah-md5-hmac ESP-Des ! crypto map MYMAP local-
address Ethernet0 crypto map MYMAP 10 ipsec-isakmp set
peer 19.19.19.19 !--- Identifies the IP address for the
destination peer router. In this case, !--- the Ethernet
interface of the remote cable modem (uBR924-1) is used.
set transform-set TUNNELSET match address 101 ! ! ! !
interface Ethernet0 ip address 18.18.18.18 255.255.255.0
ip rip send version 2 ip rip receive version 2 !
interface cable-modem0 ip rip send version 2 ip rip
receive version 2 no keepalive cable-modem downstream
saved channel 555000000 42 1 cable-modem Mac-timer t2
40000 no cable-modem compliant bridge crypto map MYMAP !
router rip version 2 network 18.0.0.0 network 172.16.0.0
! ip default-gateway 172.16.30.1 ip classless no ip http
server ! access-list 101 permit ip 18.18.18.0 0.0.0.255
19.19.19.0 0.0.0.255 snmp-server manager ! line con 0
transport input none line vty 0 4 password ww login !
end
```

CMTS uBR7246VXR也运行路由信息协议(RIP)版本2，因此路由工作。这是在CMTS使用的RIP配置：

uBR7246VXR

```
router rip
 version 2
 network 172.16.0.0
 no auto-summary
```

验证

使用本部分可确认配置能否正常运行。

为了验证IPsec工作：

- 验证这些事：Cisco IOS软件软件支持IPsec。运行的配置正确。接口是UP。路由工作。定义的访问列表加密流量正确。
- 创建流量并且查看增加的加密和解密，发现数量。
- 打开crypto的调试。

命令输出解释程序（仅限注册用户）(OIT) 支持某些 **show** 命令。使用 OIT 可查看对 show 命令输出的分析。

发出**show version命令**在两电缆调制解调器。

```
ubr924-1#show version Cisco Internetwork Operating System Software IOS (tm) 920 Software
(UBR920-K1O3SV4Y556I-M), Version 12.1(6), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by
Cisco Systems, Inc. Compiled Wed 27-Dec-00 16:36 by kellythw Image text-base: 0x800100A0, data-
base: 0x806C1C20 ROM: System Bootstrap, Version 12.0(6r)T3, RELEASE SOFTWARE (fc1) ubr924-1
uptime is 1 hour, 47 minutes System returned to ROM by reload at 10:39:05 - Fri Feb 9 2001
System restarted at 10:40:05 - Fri Feb 9 2001 System image file is "flash:ubr920-k1o3sv4y556i-
mz.121-6" cisco uBR920 CM (MPC850) processor (revision 3.e) with 15872K/1024K bytes of memory.
Processor board ID FAA0422Q04F Bridging software. 1 Ethernet/IEEE 802.3 interface(s) 1 Cable
Modem network interface(s) 3968K bytes of processor board System flash (Read/Write) 1536K bytes
of processor board Boot flash (Read/Write) Configuration register is 0x2102
```

uBR924-1运行Cisco IOS软件版本12.1(6)以VALUE SMALL OFFICE/VOICE/FW IPSEC 56特性组。

```
ubr904-2#show version Cisco Internetwork Operating System Software IOS (TM) 900 Software
(UBR900-K1OY556I-M), Version 12.1(6), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2000 by cisco
Systems, Inc. Compiled Wed 27-DEC-00 11:06 by kellythw Image text-base: 0x08004000, database:
0x085714DC ROM: System Bootstrap, Version 11.2(19980518:195057), RELEASED SOFTWARE ROM: 900
Software (UBR900-RBOOT-M), Version 11.3(11)NA, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1) ubr904-2
uptime is 1 hour, 48 minutes System returned to ROM by reload at 10:38:44 - Fri Feb 9 2001
System restarted at 10:40:37 - Fri Feb 9 2001 System image file is "flash:ubr900-k1oy556i-
mz.121-6" cisco uBR900 CM (68360) processor (revision D) with 8192K bytes of memory. Processor
board ID FAA0235Q0ZS Bridging software. 1 Ethernet/IEEE 802.3 interface(s) 1 Cable Modem network
interface(s) 4096K bytes of processor board System flash (Read/Write) 2048K bytes of processor
board Boot flash (Read/Write) Configuration register is 0x2102
```

uBR904-2运行Cisco IOS软件版本12.1(6)以小OFFICE/FW IPSec56特性组。

```
ubr924-1#show ip interface brief Interface IP-Address OK? Method Status Protocol Ethernet0
19.19.19.19 YES NVRAM up up cable-modem0 172.16.31.20 YES unset up up ubr904-2#show ip interface
brief Interface IP-Address OK? Method Status Protocol Ethernet0 18.18.18.18 YES NVRAM up up
cable-modem0 172.16.30.18 YES unset up up
```

从最后命令，您能看到以太网接口是UP。以太网接口的IP地址手工被输入。电缆接口也上，并且他们通过DHCP了解他们的IP地址。由于这些电缆地址动态地分配，他们不可能使用作为对等体在IPSec配置里。

```
ubr924-1#show ip 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 - ISIS level-1, L2 - ISIS level-2, ia - ISIS inter area * - candidate default, U -
per-user static route, o - ODR P - periodic downloaded static route Gateway of last resort is
172.16.31.1 to network 0.0.0.0 19.0.0.0/24 is subnetted, 1 subnets C 19.19.19.0 is directly
connected, Ethernet0 R 18.0.0.0/8 [120/2] via 172.16.31.1, 00:00:23, cable-modem0 172.16.0.0/16
is variably subnetted, 4 subnets, 3 masks R 172.16.135.0/25 [120/1] via 172.16.31.1, 00:00:23,
```

```
cable-modem0 R 172.16.29.0/27 [120/1] via 172.16.31.1, 00:00:23, cable-modem0 R 172.16.30.0/24
[120/1] via 172.16.31.1, 00:00:23, cable-modem0 C 172.16.31.0/24 is directly connected, cable-
modem0 R 192.168.99.0/24 [120/3] via 172.16.31.1, 00:00:24, cable-modem0 10.0.0.0/24 is
subnetted, 2 subnets R 10.10.10.0 [120/2] via 172.16.31.1, 00:00:24, cable-modem0 S* 0.0.0.0/0
[1/0] via 172.16.31.1
```

您能从此看到输出uBR924-1学习关于路由18.18.18.0，是uBR904-2以太网接口。

```
ubr904-2#show ip 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 -
ISIS, L1 - ISIS level-1, L2 - ISIS level-2, IA - ISIS inter area * - candidate default, U - per-
user static route, o - ODR P - periodic downloaded static route Gateway of last resort is
172.16.30.1 to network 0.0.0.0 R 19.0.0.0/8 [120/2] via 172.16.30.1, 00:00:17, cable-modem0
18.0.0.0/24 is subnetted, 1 subnets C 18.18.18.0 is directly connected, Ethernet0 172.16.0.0/16
is variably subnetted, 4 subnets, 3 masks R 172.16.135.0/25 [120/1] via 172.16.30.1, 00:00:17,
cable-modem0 R 172.16.29.224/27 [120/1] via 172.16.30.1, 00:00:17, cable-modem0 C 172.16.30.0/24
is directly connected, cable-modem0 R 172.16.31.0/24 [120/1] via 172.16.30.1, 00:00:17, cable-
modem0 R 192.168.99.0/24 [120/3] via 172.16.30.1, 00:00:18, cable-modem0 10.0.0.0/24 is
subnetted, 1 subnets R 10.10.10.0 [120/2] via 172.16.30.1, 00:00:18, cable-modem0 S* 0.0.0.0/0
[1/0] via 172.16.30.1
```

从uBR904-2路由表，您能看到uBR924-1以太网的网络在路由表里。

注意： 也许有您不能运行路由协议在两电缆调制解调器之间的案件。在这类情况下，您必须添加在CMTS的静态路由到电缆调制解调器的以太网接口的直接数据流。

检查的下件事是访问列表的证明;发出**show access-lists命令**在两路由器。

```
ubr924-1#show access-lists Extended IP access list 101 permit ip 19.19.19.0 0.0.0.255 18.18.18.0
0.0.0.255 (2045 matches) ubr904-2#show access-lists Extended IP access list 101 permit ip
18.18.18.0 0.0.0.255 19.19.19.0 0.0.0.255 (2059 matches)
```

访问列表设置IPSec会话，当在uBR924-1 (19.19.19.0)后时的LAN发送IP数据流对在uBR904-2 (18.18.18.0)后的LAN，反之亦然。因为它制造问题，请勿使用“其中任一”在访问列表。欲了解更详细的信息参考配置IPSec网络安全。

没有IPSec数据流。发出**show crypto engine connection active命令**。

```
ubr924-1#show crypto engine connection active ID Interface IP-Address State Algorithm Encrypt
Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 ubr904-2#show crypto engine connection active ID Interface
IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0
```

因为流量未匹配访问列表，没有IPSec连接。

注意： 使用 **debug** 命令之前，请参阅有关 Debug 命令的重要信息。

下一步是打开一些crypto调试生成关注数据流。

在本例中，这些调试打开：

- **debug crypto engine**
- **debug crypto ipsec**
- **debug crypto key-exchange**
- **debug crypto isakmp**

您必须首先生成若干关注数据流发现调试的输出。发出从uBR904-2以太网端口的一扩展ping到在uBR924-1 (IP地址19.19.19.1)的PC。

```
ubr904-2#ping ip Target IP address: 19.19.19.1 !--- IP address of PC1 behind the Ethernet of
uBR924-1. Repeat count [5]: 100 !--- Sends 100 pings. Datagram size [100]: Timeout in seconds
```

[2]: Extended commands [n]: y **Source address or interface: 18.18.18.18** *!--- IP address of the Ethernet behind uBR904-2.* Type of service [0]: Set DF bit in IP header? [no]: Validate reply data? [no]: Data pattern [0xABCD]: Loose, Strict, Record, Timestamp, Verbose[none]: Sweep range of sizes [n]: Type escape sequence to abort. Sending 100, 100-byte ICMP Echos to 19.19.19.1, timeout is 2 seconds:

uBR924-2显示此debug输出：

ubr904-2#
01:50:37: **IPSec(sa_request)**: , (key eng. msg.) **src= 18.18.18.18, dest= 19.19.19.19**, src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), protocol= AH, transform= ah-md5-hmac , lifedur= 3600s and 4608000kb, spi= 0x19911A16(428939798), conn_id= 0, keysize= 0, flags= 0x4004 01:50:37: **IPSec(sa_request):** , (key Eng. msg.) **src= 18.18.18.18, dest= 19.19.19.19**, src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= ESP-Des , lifedur= 3600s and 4608000kb, spi= 0x7091981(118036865), conn_id= 0, keysize= 0, flags= 0x4004 01:50:37: ISAKMP: received ke message (1/2) 01:50:37: ISAKMP (0:1): sitting IDLE. Starting QM immediately (QM_IDLE) 01:50:37: ISAKMP (0:1): beginning Quick Mode exchange, M-ID of 1108017901 01:50:37: CryptoEngine0: generate hmac context for conn id 1 01:50:37: ISAKMP (1): sending packet to 19.19.19.19 (I) QM_IDLE 01:50:37: ISAKMP (1): received packet from 19.19.19.19 (I) QM_IDLE 01:50:37: CryptoEngine0: generate hmac context for conn id 1 01:50:37: ISAKMP (0:1): processing SA payload. message ID = 1108017901 01:50:37: ISAKMP (0:1): Checking IPSec proposal 1 01:50:37: ISAKMP: transform 1, AH_MD5 01:50:37: ISAKMP: attributes in transform: 01:50:3.**!!!!!!!!!!!!!!!!!!!!!!!**7: ISAKMP: encaps is 1 01:50:37: ISAKMP: SA life type in seconds 01:50:37: ISAKMP: SA life duration (basic) of 3600 01:50:37: ISAKMP: SA life type in kilobytes 01:50:37: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 01:50:37: ISAKMP: authenticator is HMAC-MD5 01:50:37: validate proposal 0 01:50:37: ISAKMP (0:1): atts are acceptable. 01:50:37: ISAKMP (0:1): Checking IPSec proposal 1 01:50:37: ISAKMP: transform 1, ESP_DES 01:50:37: ISAKMP: attributes in transform: 01:50:37: ISAKMP: encaps is 1 01:50:37: ISAKMP: SA life type in seconds 01:50:37: ISAKMP: SA life duration (basic) of 3600 01:50:37: ISAKMP: SA life type in kilobytes 01:50:37: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 01:50:37: validate proposal 0 01:50:37: ISAKMP (0:1): atts are acceptable. 01:50:37: IPSec(validate_proposal_request): proposal part #1, (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18, dest_proxy= 19.19.1**!!! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Success rate is 99 percent** (99/100), round-trip min/avg/max = 30/40/70 ms ubr904-2#

注意第一次ping失败。这是因为需要建立连接。

uBR924-1显示此debug输出：

ubr924-1#
01:50:24: **ISAKMP (1): received packet from 18.18.18.18** (R) QM_IDLE 01:50:24: CryptoEngine0: generate hmac context for conn id 1 01:50:24: ISAKMP (0:1): **processing SA payload.** Message ID = 1108017901 01:50:24: ISAKMP (0:1): Checking IPSec proposal 1 01:50:24: ISAKMP: transform 1, AH_MD5 01:50:24: ISAKMP: attributes in transform: 01:50:24: ISAKMP: encaps is 1 01:50:24: ISAKMP: SA life type in seconds 01:50:24: ISAKMP: SA life duration (basic) of 3600 01:50:24: ISAKMP: SA life type in kilobytes 01:50:24: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 01:50:24: ISAKMP: **authenticator is HMAC-MD5** 01:50:24: validate proposal 0 01:50:24: ISAKMP (0:1): atts are acceptable. 01:50:24: ISAKMP (0:1): Checking IPSec proposal 1 01:50:24: ISAKMP: **transform 1, ESP_DES** 01:50:24: ISAKMP: attributes in transform: 01:50:24: ISAKMP: encaps is 1 01:50:24: ISAKMP: SA life type in seconds 01:50:24: ISAKMP: SA life duration (basic) of 3600 01:50:24: ISAKMP: SA life type in kilobytes 01:50:24: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 01:50:24: **validate proposal 0** 01:50:24: ISAKMP (0:1): atts are acceptable. 01:50:24: IPSec**(validate_proposal_request)**: proposal part #1, (key Eng. msg.) **dest= 19.19.19.19, src= 18.18.18.18,** dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= AH, transform= ah-md5-hmac** , lifedur= 0s and 0kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4 01:50:24: IPSec(validate_proposal_request): proposal part #2, (key Eng. msg.) **dest= 19.19.19.19, src= 18.18.18.18**, dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= ESP, transform= ESP-Des** , lifedur= 0s and 0kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4 01:50:24: validate proposal request 0 01:50:24: ISAKMP (0:1): processing NONCE payload. Message ID = 1108017901 01:50:24: ISAKMP (0:1): processing ID payload. Message ID = 1108017901 01:50:24: ISAKMP (1): ID_IPV4_ADDR_SUBNET src 18.18.18.0/255.255.255.0 prot 0 Port 0 01:50:24: ISAKMP (0:1): processing ID payload. Message ID = 1108017901 01:50:24: ISAKMP (1):

ID_IPV4_ADDR_SUBNET dst 19.19.19.0/255.255.255.0 prot 0 Port 0 01:50:24: **ISAKMP (0:1): asking for 2 spis from IPSec** 01:50:24: IPSec(key_engine): got a queue event... 01:50:24: IPSec(spi_response): getting spi 393021796 for SA from 18.18.18.18 to 19.19.19.19 for prot 2 01:50:24: IPSec(spi_response): getting spi 45686884 for SA from 18.18.18.18 to 19.19.19.19 for prot 3 01:50:24: **ISAKMP: received ke message (2/2)** 01:50:24: CryptoEngine0: generate hmac context for conn id 1 01:50:24: ISAKMP (1): sending packet to 18.18.18.18 (R) QM_IDLE 01:50:24: ISAKMP (1): received packet from 18.18.18.18 (R) QM_IDLE 01:50:24: **CryptoEngine0: generate hmac context for conn id 1** 01:50:24: IPSec allocate flow 0 01:50:24: IPSec allocate flow 0 01:50:24: **ISAKMP (0:1): Creating IPSec SAs** 01:50:24: **inbound SA from 18.18.18.18 to 19.19.19.19 (proxy 18.18.18.0 to 19.19.19.0)** 01:50:24: has spi 393021796 and conn_id 2000 and flags 4 01:50:24: lifetime of 3600 seconds 01:50:24: lifetime of 4608000 kilobytes 01:50:24: **outbound SA from 19.19.19.19 to 18.18.18.18 (proxy 19.19.19.0 to 18.18.18.0)** 01:50:24: has spi 428939798 and conn_id 2001 and flags 4 01:50:24: lifetime of 3600 seconds 01:50:24: lifetime of 4608000 kilobytes 01:50:24: **ISAKMP (0:1): Creating IPSec SAs** 01:50:24: **inbound SA from 18.18.18.18 to 19.19.19.19 (proxy 18.18.18.0 to 19.19.19.0)** 01:50:24: has spi 45686884 and conn_id 2002 and flags 4 01:50:24: lifetime of 3600 seconds 01:50:24: lifetime of 4608000 kilobytes 01:50:24: **outbound SA from 19.19.19.19 to 18.18.18.18 (proxy 19.19.19.0 to 18.18.18.0)** 01:50:24: has spi 118036865 and conn_id 2003 and flags 4 01:50:25: lifetime of 3600 seconds 01:50:25: lifetime of 4608000 kilobytes 01:50:25: ISAKMP (0:1): deleting node 1108017901 error FALSE reason "quick mode done (await()" 01:50:25: **IPSec(key_engine): got a queue event...** 01:50:25: **IPSec(initialize_sas):** , (key Eng. msg.) **dest= 19.19.19.19, src= 18.18.18.18**, dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= AH, transform= ah-md5-hmac** , lifedur= 3600s and 4608000kb, spi= 0x176D0964(393021796), **conn_id= 2000**, keysize= 0, flags= 0x4 01:50:25: **IPSec(initialize_sas):** , (key Eng. msg.) **src= 19.19.19.19, dest= 18.18.18.18**, src_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), dest_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= AH, transform= ah-md5-hmac** , lifedur= 3600s and 4608000kb, spi= 0x19911A16(428939798), **conn_id= 2001**, keysize= 0, flags= 0x4 01:50:25: **IPSec(initialize_sas):** , (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18, dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= ESP, transform= ESP-Des** , lifedur= 3600s and 4608000kb, spi= 0x2B92064(45686884), **conn_id= 2002**, keysize= 0, flags= 0x4 01:50:25: **IPSec(initialize_sas):** , (key Eng. msg.) **src= 19.19.19.19, dest= 18.18.18.18**, src_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4), dest_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4), **protocol= ESP, transform= ESP-Des** , lifedur= 3600s and 4608000kb, spi= 0x7091981(118036865), **conn_id= 2003**, keysize= 0, flags= 0x4 01:50:25: IPSec(create_sa): sa created, (sa) sa_dest= 19.19.19.19, sa_prot= 51, sa_spi= 0x176D0964(393021796), sa_trans= ah-md5-hmac , sa_conn_id= 2000 01:50:25: IPSec(create_sa): sa created, (sa) sa_dest= 18.18.18.18, sa_prot= 51, sa_spi= 0x19911A16(428939798), sa_trans= ah-md5-hmac , sa_conn_id= 2001 01:50:25: IPSec(create_sa): sa created, (sa) sa_dest= 19.19.19.19, sa_prot= 50, sa_spi= 0x2B92064(45686884), sa_trans= ESP-Des , sa_conn_id= 2002 01:50:25: IPSec(create_sa): sa created, (sa) sa_dest= 18.18.18.18, sa_prot= 50, sa_spi= 0x7091981(118036865), sa_trans= ESP-Des , sa_conn_id= 2003 ubr924-1#

一旦IPSec隧道创建，您能看到连接和加密和解密的信息包。

ubr924-1#**show crypto engine connection active** ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 cable-modem0 172.16.31.20 set **HMAC_MD5 0 99** 2001 cable-modem0 172.16.31.20 set HMAC_MD5 **99 0** 2002 cable-modem0 172.16.31.20 set **DES_56_CBC 0 99** 2003 cable-modem0 172.16.31.20 set DES_56_CBC **99 0**

第一200x行显示接收的99数据包。它必须解码数据包为了发送他们到PC1。第二行显示99被发送的数据包。在发送他们对uBR904-2前，它必须加密数据包。第三条和第四条线路执行同一进程，但是与ESP-DES转换而不是AH-MD5-HMAC。

注意： 如果在有线调制解调器配置的转换设置是ESP-DES ESP-MD5-HMAC，您只看到两自治系统(AS)，与在上一个**show命令**显示的四相对。

ubr904-2#**show crypto engine connection active** ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 **cable-modem0 172.16.30.18** set **HMAC_MD5 0 99** 2001 cable-modem0 172.16.30.18 set HMAC_MD5 **99 0** 2002 **cable-modem0 172.16.30.18** set **DES_56_CBC 0 99** 2003 cable-modem0 172.16.30.18 set DES_56_CBC **99 0**

发出扩展ping对从uBR924-1的PC2发现计数器是否为加密和解密的信息包增加。

ubr924-1#**ping ip Target IP address: 18.18.18.1** Repeat count [5]: **50** Datagram size [100]: Timeout

```
in seconds [2]: Extended commands [n]: y Source address or interface: 19.19.19.19 Type of service [0]: Set DF bit in IP header? [no]: Validate reply data? [no]: Data pattern [0xABCD]: Loose, Strict, Record, Timestamp, Verbose[none]: Sweep range of sizes [n]: Type escape sequence to abort. Sending 50, 100-byte ICMP Echos to 18.18.18.1, timeout is 2 seconds: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Success rate is 100 percent (50/50), round-trip min/avg/max = 28/30/33 ms ubr924-1#show crypto engine connection active ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 cable-modem0 172.16.31.20 set HMAC_MD5 0 149 2001 cable-modem0 172.16.31.20 set HMAC_MD5 149 0 2002 cable-modem0 172.16.31.20 set DES_56_CBC 0 149 2003 cable-modem0 172.16.31.20 set DES_56_CBC 149 0 ubr904-2#show crypto engine connection active ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 cable-modem0 172.16.30.18 set HMAC_MD5 0 149 2001 cable-modem0 172.16.30.18 set HMAC_MD5 149 0 2002 cable-modem0 172.16.30.18 set DES_56_CBC 0 149 2003 cable-modem0 172.16.30.18 set DES_56_CBC 149 0
```

另一扩展ping可以发出，发现计数器再增加。这时，发送从uBR904-2的一500数据包ping到以太网接口uBR924-1 (19.19.19.19)。

```
ubr904-2#ping ip Target IP address: 19.19.19.19 Repeat count [5]: 500 Datagram size [100]: 1000 Timeout in seconds [2]: Extended commands [n]: y Source address or interface: 18.18.18.18 Type of service [0]: Set DF bit in IP header? [no]: Validate reply data? [no]: Data pattern [0xABCD]: Loose, Strict, Record, Timestamp, Verbose[none]: Sweep range of sizes [n]: Type escape sequence to abort. Sending 500, 1000-byte ICMP Echos to 19.19.19.19, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! 01:59:06: IPSec(encapsulate): encaps area too small, moving to new buffer: idbtype 0, encaps_size 26, header size 60, avail 84!!!!!!!!! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!!!!!!!! Success rate is 100 percent (500/500), round-trip min/avg/max = 98/135/352 ms ubr904-2#show crypto engine connection active ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 cable-modem0 172.16.30.18 set HMAC_MD5 0 649 2001 cable-modem0 172.16.30.18 set HMAC_MD5 649 0 2002 cable-modem0 172.16.30.18 set DES_56_CBC 0 649 2003 cable-modem0 172.16.30.18 set DES_56_CBC 649 0 ubr924-1#show crypto engine connection active ID Interface IP-Address State Algorithm Encrypt Decrypt 1 set HMAC_MD5+DES_56_CB 0 0 2000 cable-modem0 172.16.31.20 set HMAC_MD5 0 649 2001 cable-modem0 172.16.31.20 set HMAC_MD5 649 0 2002 cable-modem0 172.16.31.20 set DES_56_CBC 0 649 2003 cable-modem0 172.16.31.20 set DES_56_CBC 649 0
```

您能发出**clear crypto isakmp**，并且**clear crypto sa**发出命令清除连接。并且，如果在有效期，没有在IPSec隧道间的流量， IPsec重置自动连接。

故障排除

当前没有故障排除此配置的特定可用资料。

相关信息

- IPSec网络安全命令
- 一IP安全-调试信息
- IPSec 配置示例
- 配置 IPSec 网络安全
- 配置Cisco uBR900系列电缆接入路由器
- Cisco 有线/宽频下载(注册用户)
- 宽带有线支持
- 技术支持和文档 - Cisco Systems