

IPSec au-dessus des configurations et des debugs d'échantillon de câble

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Théorie générale](#)

[Configurez](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérifiez](#)

[Dépannez](#)

[Informations connexes](#)

[Introduction](#)

L'IPSec (IPsec) est un cadre des standards ouverts qui assure les communications privées sécurisées au-dessus des réseaux IP. Basé sur des normes développées par l'Internet Engineering Task Force (IETF), IPsec assure la confidentialité, l'intégrité, et l'authenticité des communications de données à travers un réseau IP public. IPsec fournit un composant nécessaire pour un basé sur des standards, solution flexible pour déployer une stratégie de sécurité à l'échelle du réseau entier.

Ce document fournit un exemple de configuration d'IPsec entre deux modems câble Cisco. Cette configuration crée un tunnel de chiffrement à travers un réseau câblé entre deux Routeurs de modem câblé de gamme Cisco uBR9xx. Tout le trafic entre les deux réseaux est chiffré. Mais on permet au le trafic destiné pour d'autres réseaux pour passer décrypté. Pour le petit bureau, les utilisateurs du bureau à domicile (SOHO), ceci permet la création du Réseaux privés virtuels (VPN) à travers un réseau câblé.

[Conditions préalables](#)

[Conditions requises](#)

Aucune spécification déterminée n'est requise pour ce document.

[Composants utilisés](#)

Les Modems doivent répondre à ces exigences de configurer IPsec sur deux Modems câble :

- Cisco uBR904, uBR905, ou uBR924 en mode de routage
- Ensemble de caractéristiques d'IPsec 56
- Version de logiciel 12.0(5)T ou ultérieures de Cisco IOS®

En outre, vous devez avoir un système de terminaison par modem câble (CMTS), qui est tout Data-over-Cable Service Interface Specifications (DOCSIS) - routeur câble conforme de headend, tel que Cisco uBR7246, Cisco uBR7223, ou Cisco uBR7246VXR.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

Théorie générale

L'exemple dans ce document utilise un modem câble uBR904, un modem câble uBR924, et un uBR7246VXR CMTS. Les Modems câble exécutent le Logiciel Cisco IOS version 12.1(6), et le CMTS exécute la version du logiciel Cisco IOS 12.1(4)EC.

Remarque: Cet exemple est fait avec la configuration manuelle sur les Modems câble par le port de console. Si un traitement automatisé est exécuté par le fichier de configuration DOCSIS (le script ios.cfg est créé avec les Listes d'accès de configuration d'IPsec) puis 100 et 101 ne peut pas être utilisé. C'est parce que l'implémentation de Cisco de la table de docsDevNmAccess de Protocole SNMP (Simple Network Management Protocol) utilise des Listes d'accès de Cisco IOS. Il crée une liste d'accès par interface. Sur uBR904, 924, et 905, les deux premières Listes d'accès sont généralement utilisés (100 et 101). Sur un modem câble qui prend en charge le bus USB (USB), comme le CVA120, trois Listes d'accès sont utilisés (100, 101, et 102).

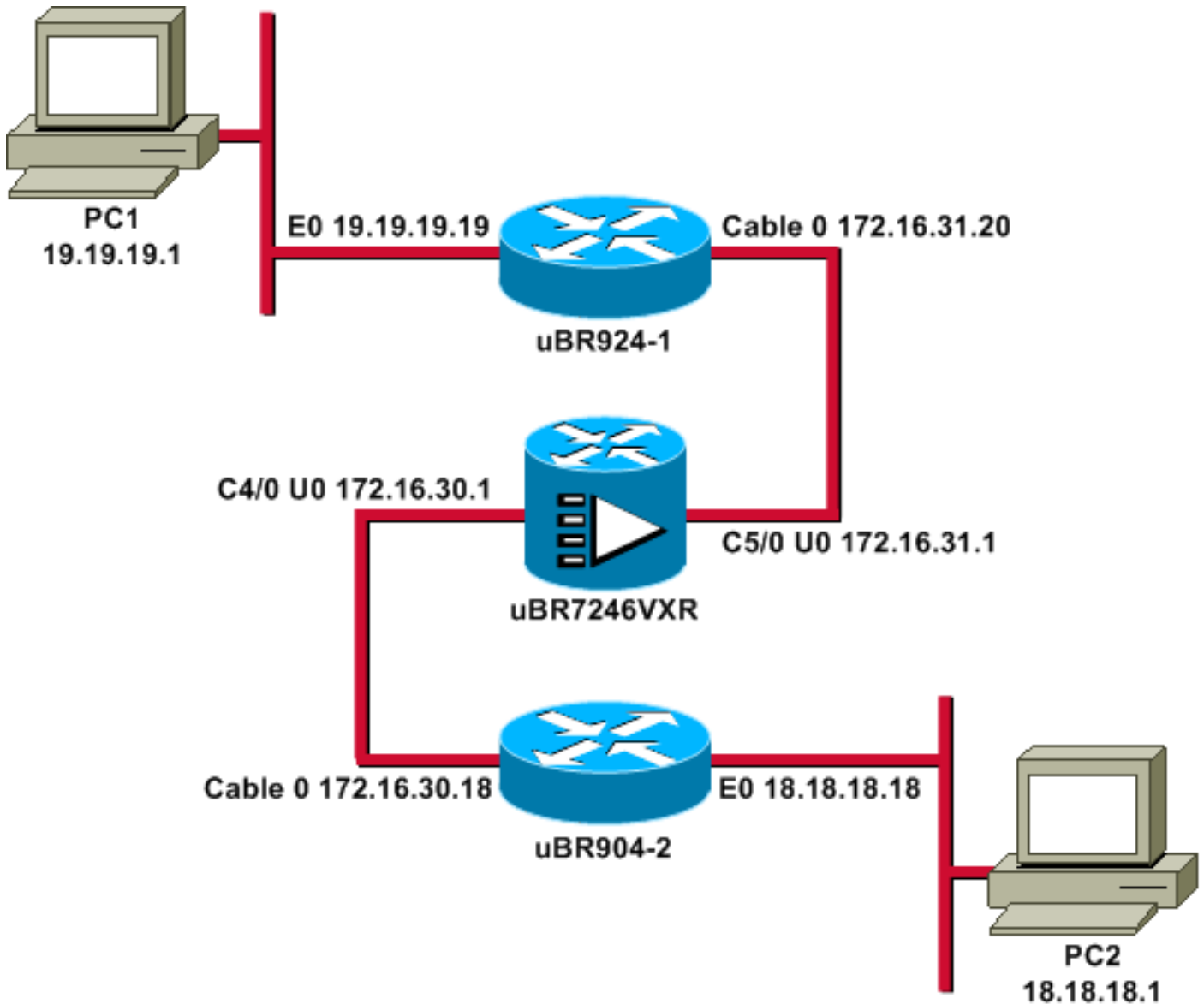
Configurez

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

Remarque: Utilisez le [Command Lookup Tool](#) (clients [enregistrés](#) seulement) pour trouver les informations complémentaires au sujet des commandes dans ce document.

Diagramme du réseau

Ce document utilise la configuration réseau suivante :



Remarque: Toutes les adresses IP dans ce diagramme ont un masque 24-bit.

Configurations

Ce document utilise les configurations suivantes :

- [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
```

La configuration de l'autre modem câble est très semblable, ainsi la plupart des commentaires dans la configuration précédente sont omises.

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
```

Le CMTS uBR7246VXR exécute également la version 2 de Protocole RIP (Routing Information Protocol), de sorte que le routage fonctionne. C'est la configuration RIP utilisée sur le CMTS :

uBR7246VXR

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

Vérifiez

Référez-vous à cette section pour vous assurer du bon fonctionnement de votre configuration.

Afin de vérifier qu'IPsec fonctionne :

- Vérifiez ces choses : Les supports logiciels IPsec de Cisco IOS. La configuration en cours est correcte. Les interfaces sont en hausse. Acheminement des travaux. La liste d'accès définie pour chiffrer le trafic est correcte.
- Créez le trafic et regardez le chiffrer et le déchiffrement, pour voir la quantité qui augmente.
- Activez met au point pour crypto.

L'[Outil Interpréteur de sortie](#) (clients [enregistrés](#) uniquement) (OIT) prend en charge certaines commandes **show**. Utilisez l'OIT pour afficher une analyse de la sortie de la commande **show**.

Émettez la commande de **show version** sur les deux Modems câble.

```
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
```

L'uBR924-1 exécute le Logiciel Cisco IOS version 12.1(6) avec le PETIT OFFICE/VOICE/FW positionnement de caractéristique d'IPSec 56 de la VALEUR.

```
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
```

L'uBR904-2 exécute le Logiciel Cisco IOS version 12.1(6) avec le PETIT positionnement de caractéristique d'IPSec 56 OFFICE/FW.

```
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
```

De la dernière commande, vous pouvez voir que les interfaces Ethernet sont en hausse. Les adresses IP des interfaces Ethernet ont été manuellement écrites. Les interfaces de câble sont également hautes et elles ont appris leurs adresses IP par le DHCP. Puisque ces adresses de câble sont dynamiquement assignées, elles ne peuvent pas être utilisées comme pairs en [configuration IPSec](#).

```
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

Vous pouvez voir du ce pour sortir qu'uBR924-1 apprend au sujet de l'artère 18.18.18.0, qui est l'interface Ethernet d'uBR904-2.

```
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
```

De la table de routage d'uBR904-2, vous pouvez voir que le réseau pour les Ethernets d'uBR924-1 est dans la table de routage.

Remarque: Il pourrait y avoir des cas où vous ne pouvez pas exécuter un protocole de routage entre les deux Modems câble. En pareil cas, vous devez ajouter les artères statiques sur le CMTS pour se diriger le trafic pour les interfaces Ethernet des Modems câble.

La prochaine chose à vérifier est la certification de la liste d'accès ; émettez la commande de **show access-lists** sur les deux Routeurs.

```
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)
```

La liste d'accès a placé la session d'IPsec quand le RÉSEAU LOCAL derrière uBR924-1 (19.19.19.0) envoie le trafic IP au RÉSEAU LOCAL derrière uBR904-2 (18.18.18.0), et vice versa. N'en utilisez pas « » sur les Listes d'accès, parce qu'il crée des problèmes. Référez-vous à la [sécurité des réseaux de configuration d'IPSec](#) pour plus de détails.

Il n'y a aucun trafic d'IPsec. Émettez la commande de **show crypto engine connection active**.

```
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
```

Il n'y a aucune connexion d'IPsec parce qu'aucun trafic n'a apparié les Listes d'accès.

Remarque: Référez-vous aux [informations importantes sur les commandes de débogage](#) avant d'utiliser les commandes de débogage.

L'étape suivante est d'activer un certain crypto met au point pour générer le trafic intéressant.

Dans cet exemple, ceux-ci met au point sont activés :


```

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#

```

Une fois que le tunnel d'IPsec est créé, vous pouvez voir la connexion et les paquets chiffrés et déchiffrés.

```

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

```

La première ligne 200x affiche les 99 paquets reçus. Il doit déchiffrer les paquets afin de les envoyer à PC1. La deuxième ligne affiche 99 paquets envoyés. Il doit chiffrer les paquets avant

qu'il les envoie à uBR904-2. Les troisième et quatrième lignes font le même processus, mais avec ESP-DES le transforment au lieu d'AH-MD5-HMAC.

Remarque: Si le jeu de transformations qui est configuré sur le modem câble est ESP-DES ESP-MD5-HMAC, vous voyez seulement deux systèmes autonomes (AS), par opposition aux quatre affichés dans la **commande show** précédente.

```
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
```

Fournissez un ping étendu à PC2 d'uBR924-1 pour voir si les compteurs incrémentsent pour les paquets chiffrés et déchiffrés.

```
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
```

Un autre ping étendu peut être émis, pour voir que les compteurs incrémentsent de nouveau. Cette fois, envoient un ping 500-packet d'uBR904-2 à l'interface Ethernet d'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
```

Vous pouvez émettre le **clear crypto isakmp** et le **clear crypto sa** commande d'effacer les connexions. En outre, s'il n'y a aucun trafic à travers le tunnel d'IPsec pendant le temps d'expiration, IPsec remet à l'état initial la connexion automatiquement.

[**Dépannez**](#)

Il n'y a actuellement aucune informations disponibles spécifique pour dépanner cette configuration.

[Informations connexes](#)

- [Commandes de Sécurité de réseau IPSec](#)
- [Une introduction au cryptage de sécurité IP \(IPsec\) - les informations de debug](#)
- [Exemples de configuration d'IPsec](#)
- [Sécurité des réseaux de configuration d'IPSec](#)
- [Configuration du Routeurs d'accès par câble de la gamme Cisco uBR900](#)
- [Téléchargements de câble Cisco/Large bande \(clients \[enregistrés\]\(#\) seulement\)](#)
- [Support pour la technologie de câble haut débit](#)
- [Support et documentation techniques - Cisco Systems](#)