Présentation du chiffrement IPSec (IP Security)

Contenu

Introduction

Ce document présente IPsec aux utilisateurs dans un format rapide et concis. Ce document contient des configurations de base de l'Échange de clés Internet (IKE) avec des clés pré-partagées, IKE avec une Autorité de certification et IPsec. Ce n'est pas un document approfondi. Cela dit, ce document vous permet de comprendre les tâches et l'ordre dans lequel elles sont accomplies.

 **Avertissement :** Il existe des restrictions importantes à l'exportation d'un chiffrement renforcé. Si vous violez la loi fédérale des États-Unis, alors vous, pas Cisco, n'êtes jugé responsable. Si vous avez des questions liées au contrôle d'exportation, envoyez un email à export@cisco.com.

Remarque: Multicast et Broadcast ne sont pas pris en charge sur le LAN normal ou les tunnels LAN, ou encore sur les clients VPN qui se terminent sur tous les périphériques. Le Multicast peut être passé seulement sur des tunnels GRE. Il est pris en charge seulement sur des routeurs et

pas sur les concentrateurs VPN 3000 ou les pare-feu (ASA/PIX).

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Conventions

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

Fond

IPsec est plate-forme de la deuxième génération de couche réseau la crypto pour les Plateformes de sécurité Cisco (logiciel de Cisco IOS®, PIX, et ainsi de suite). Initialement décrit dans les RFC 1825 à 1829, qui sont maintenant obsolètes, IPsec est actuellement abordé dans un certain nombre de documents présentés par le Groupe de travail sur la sécurité IP IETF. IPsec prend actuellement en charge les paquets de monodiffusion 4 version IP. La prise en charge de la multidiffusion et IPv6 aura lieu ultérieurement.

Les points forts d'IPsec par rapport aux offres de chiffrement Cisco actuelles sont les suivants :

1. **Pluri-constructeurs** - Puisque le cadre d'IPsec est normalisé, les clients ne sont verrouillés dans aucun produit spécifique de constructeur. IPsec est présent sur les routeurs, les pare-feu et clients et les ordinateurs clients (Windows, Mac, etc.).
2. **Évolutivité** - IPsec est conçu pour les grandes entreprises. Par conséquent, il inclut la gestion de clés intégrée.

Remarque: Tandis que plusieurs plates-formes Cisco peuvent utiliser IPsec, ce document est orienté sur le logiciel Cisco IOS.

Jargon chiffrement (Vocabulaire)

Vous devez connaître ces termes afin de comprendre IPsec et lire le reste de ce document. Quand vous voyez des acronymes dans d'autres parties de ce document, référez-vous à cette page pour obtenir des définitions.

Advanced Encryption Standard (AES) - AES a été finalisé comme algorithme de chiffrement approuvé selon la norme FIPS (Federal Information Processing Standard) à utiliser afin de protéger la transmission de données électroniques (FIPS PUB 197). AES est basé sur l'algorithme de Rijndael, qui spécifie comment employer des clés avec une longueur de 128, 192, ou 256 bits pour chiffrer des blocs avec une longueur de 128, 192, ou 256 bits. Chacune des neuf combinaisons de longueur de clé et de longueur de bloc est possible.

Authentication Header (AH) - Protocole de sécurité qui fournit des services d'authentification et des services facultatifs de relecture-détection. AH est intégré dans les données à protéger, par exemple, un datagramme IP complet. AH peut être utilisé tout seul ou avec ESP (Encryption Service Payload). Référez-vous au RFC 2402.

Authentification - Une des fonctions du cadre d'IPsec. L'authentification établit l'intégrité du flux de données et s'assure qu'il n'est pas falsifié en transit. Elle fournit également la confirmation au sujet de l'origine du flux de données.

Autorité de certification (CA) - Entité tiers responsable de délivrer et d'annuler des certificats. Chaque périphérique qui a son propre certificat et sa propre clé publique de l'Autorité de certification peut authentifier tous les autres périphériques dans un domaine de CA donné. Ce terme s'applique également au logiciel du serveur qui fournit ces services.

Certificat - Objet signé par chiffrement qui contient une identité et une clé publique associée avec cette identité.

Chiffrement classique - Mécanisme de chiffrement propriétaire Cisco utilisé dans le Logiciel Cisco IOS Version 11.2. Le chiffrement classique est disponible dans le Logiciel Cisco IOS Version 11.3. Cependant, IPsec n'est pas fourni au Logiciel Cisco IOS Version 11.2. Les documents marketing font également référence au chiffrement classique de nom sous l'appellation Encryption Express ou Cisco Encryption Technology (CET).

Liste des révocations de certificat (CRL) - Message signé numériquement qui répertorie tous les certificats révoqués mentionnés par une CA donnée. Cela ressemble à un carnet de numéros de cartes bancaires volées qui permet aux magasins de refuser des cartes de crédit non fiables.

Carte de chiffrement - Entité de configuration de logiciel Cisco IOS qui remplit deux fonctions principales. D'abord, elle sélectionne les flux de données qui ont besoin d'être sécurisés. Ensuite, elle définit la politique pour ces flux et l'homologue de chiffrement vers lequel le trafic doit aller.

Une carte de chiffrement est appliquée à une interface. Le concept de carte de chiffrement a été introduit dans le chiffrement classique mais a été étendu pour IPsec.

Intégrité des données - Mécanismes d'intégrité des données, via l'utilisation d'algorithmes basés sur des clés secrètes ou publiques, qui permettent au destinataire d'une portion de données protégées de vérifier que les données n'ont pas été modifiées en transit.

Confidentialité des données - Méthode de manipulation des données qui empêche un attaquant de les lire. Cela est généralement assuré par le chiffrement des données et les clés qui sont seulement disponibles pour les parties impliquées dans la transmission.

Authentification de l'origine des données - Service de sécurité où le destinataire peut vérifier que les données protégées proviennent uniquement de l'expéditeur. Ce service exige un service d'intégrité des données assorti d'un mécanisme de distribution de clés, où une clé secrète est partagée seulement par l'expéditeur et le récepteur.

Data Encryption Standard (DES) - Le DES a été édité en 1977 par le National Bureau of Standards et est un dispositif de chiffrement de clé secrète basé sur l'algorithme Lucifer d'IBM. Le contraste du DES est la clé publique. Cisco utilise le DES en chiffrement classique (longueurs de clé 40 bits et 56 bits), le chiffrement IPsec (clé 56 bits), et sur le pare-feu PIX (clé 56 bits).

Diffie-Hellman - Méthode d'établissement d'une clé partagée sur un média non sécurisé. Diffie-

Hellman est un composant d'Oakley, qui est défini dans cette liste de définition.

DSS — Un algorithme de signature numérique a conçu par le National Institute of Standards and Technology des USA (NIST) basé sur la cryptographie à clé publique. Le DSS ne traite pas le chiffrement du datagramme utilisateur. Le DSS est un composant en chiffrement classique, comme la carte IPsec Redcreek, mais pas dans l'IPsec implémenté dans le logiciel Cisco IOS.

Adaptateur de service de chiffrement (ESA) - Accélérateur de chiffrement matériel utilisé dans :

- Routeurs Cisco 7204 et 7206
- Versatile Interface Processor2 (VIP2-40s) de deuxième génération dans tous les routeurs de la gamme Cisco 7500
- VIP2-40 dans les routeurs de la gamme Cisco 7000 qui sont dotés du processeur de commutation routage de la gamme Cisco 7000 (RSP7000) et des cartes d'interface du châssis de la gamme Cisco 7000 (RSP7000CI).

IPsec n'utilise pas l'accélération ESA, mais fonctionne dans un boîtier qui comporte une carte ESA sur une base réservée au logiciel.

Encapsulating Security Payload (ESP) - Protocole de sécurité qui garantit la confidentialité des données et la protection avec des services facultatifs d'authentification et de relecture-détection. ESP encapsule complètement les données utilisateur. ESP peut être utilisé par lui-même ou en même temps qu'AH. Référez-vous à RFC 2406 : Encapsulating Security Payload (ESP) IP.

Hash - Fonction à sens unique qui prend un message d'entrée de longueur arbitraire et produit un résumé de longueur fixe. Cisco utilise le Secure Hash Algorithm (SHA) et le Message Digest 5 (MD5) hache dans notre mise en place du cadre d'IPsec. Consultez la définition pour HMAC pour plus d'informations.

HMAC — C'est un mécanisme pour l'authentification de message que les utilisations cryptographiques hache comme le SHA et le MD5. Référez-vous à RFC 2104 pour une discussion approfondie de HMAC.

Échange de clés Internet (IKE) - Protocole hybride qui utilise partiellement Oakley et une autre suite de protocole appelée SKEME à l'intérieur du cadre d'Internet Security Association and Key Management Protocol (ISAKMP). IKE est utilisé pour établir une politique de sécurité partagée et des clés authentifiées pour les services, tels qu'IPsec, qui nécessitent des clés. Avant que n'importe quel trafic IPsec puisse passer, chaque routeur/pare-feu/hôte doit pouvoir vérifier l'identité de son homologue. Pour ce faire, introduisez manuellement les clés pré-partagées dans les deux hôtes, par le biais d'un service de CA, ou le DNS sécurisé à venir (DNSSec). Protocole connu autrefois sous le nom d'ISAKMP/Oakley, et qui est défini dans RFC 2409 : Échange de clés Internet (IKE)). Un élément potentiel de confusion est que les acronymes ISAKMP et IKE sont tous deux utilisés dans le logiciel Cisco IOS afin de se rapporter à la même chose. Ces deux éléments sont quelque peu différents.

Internet Security Association and Key Management Protocol (ISAKMP) - Cadre de protocole qui définit les mécanismes de l'implémentation d'un protocole d'échange de clés et de négociation d'une politique de sécurité. ISAKMP est défini dans l'Internet Security Association and Key Management Protocol (ISAKMP).

Transparence NAT d'IPsec - La fonctionnalité de transparence NAT d'IPsec introduit la prise en charge du trafic de sécurité IP (IPsec) pour voyager via les points de Traduction d'adresses de réseau (NAT) ou de Traduction d'adresses de points (PAT) dans le réseau en corrigeant de

nombreuses incompatibilités connues entre NAT et IPsec. Le NAT Traversal est une fonctionnalité qui est automatiquement détectée par les périphériques VPN. Aucune étape de configuration existe pour un routeur qui exécute le Logiciel Cisco IOS Version 12.2(13)T et ultérieur. Si les deux périphériques VPN sont NAT-T, NAT Traversal est détecté et négocié automatiquement.

ISAKMP/Oakley - Voir IKE.

Message Digest 5 (MD5) - Algorithme de hachage à sens unique qui produit un hachage 128 bits. MD5 et le Secure Hash Algorithm (SHA) sont des variations sur MD4, qui est conçu pour renforcer la sécurité de cet algorithme de hachage. SHA est plus sécurisé que MD4 et MD5. Cisco utilise les hachages pour l'authentification dans le cadre d'IPsec.

Oakley - Protocole d'échange de clés qui définit comment acquérir le matériel de clé authentifié. Le mécanisme de base pour Oakley est l'algorithme d'échange de clés Diffie-Hellman. Vous pouvez rechercher la norme dans RFC 2412 : Protocole de détermination de clé OAKLEY.

Perfect Forward Secrecy (PFS) - Le PFS s'assure qu'une clé SA IPsec donnée ne dérive d'aucun autre élément secret, tel que d'autres clés. En d'autres termes, si une clé est rompue, le PFS s'assure que l'attaquant ne peut pas dériver toute autre key. Si le PFS n'est pas activé, quelqu'un peut potentiellement rompre la clé secrète SA IKE, copier toutes les données protégées IPsec, et puis employer la connaissance de l'élément secret SA IKE afin de compromettre la configuration des SA IPsec par cette SA IKE. Avec PFS, la rupture IKE ne donne pas à un attaquant l'accès immédiat à IPsec. L'attaquant doit rompre chaque SA IPsec individuellement. L'implémentation d'IPsec Cisco IOS utilise le groupe 1 PFS (D-H 768 bits) par défaut.

Relecture-détection - Service de sécurité où le récepteur peut refuser des paquets anciens ou dupliqués afin de contrer des attaques par relecture. Les attaques par relecture reposent sur l'attaquant qui envoie des paquets anciens ou dupliqués au récepteur pour faire penser à un trafic de bogue légitime. La relecture-détection s'effectue par l'utilisation de numéros de séquence combinés avec l'authentification et est une fonctionnalité standard d'IPsec.

RSA — C'est un algorithme de chiffrement de clé publique, nommé après ses inventeurs, Rivest, Shamir et Adleman, avec une longueur principale variable. La faiblesse principale de RSA est qu'il est sensiblement plus lent pour calculer que les algorithmes populaires à clé secrète, tels que DES. L'implémentation IKE Cisco utilise un échange de Diffie-Hellman afin d'obtenir les clés secrètes. Cet échange peut être authentifié avec RSA, ou des clés pré-partagées. Avec l'échange de Diffie-Hellman, la clé DES ne croise jamais le réseau, pas même sous la forme chiffrée, ce qui n'est pas le cas avec la technique de signe et de chiffrement RSA. RSA n'est pas un domaine public et doit être sous licence de la sécurité des données RSA.

Association de sécurité (SA) - Instance de politique de sécurité et de matériel de clé appliquée à un flux de données. IKE et IPsec utilisent les SA, bien qu'elles soient indépendantes les unes des autres. Les SA IPsec sont unidirectionnelles et uniques dans chaque protocole de sécurité. Un ensemble de SA sont nécessaires pour un tuyau de données protégé, un par direction par protocole. Par exemple, si vous avez un tuyau qui supporte ESP entre homologues, une SA ESP est nécessaire pour chaque direction. Les SA sont seulement identifiées par adresse de destination (point de terminaison IPsec), protocole de sécurité (AH ou ESP), et un SPI (Security Parameters Index).

IKE négocie et établit les SA au nom d'IPsec. Un utilisateur peut également établir les SA IPsec manuellement.

Une SA IKE est utilisé par IKE seulement. À la différence de la SA IPsec, elle est bidirectionnelle.

Secure Hash Algorithm (SHA) - Hachage à sens unique avancé par NIST. SHA est étroitement modelé d'après MD4 et produit un résumé 160 bits. Puisque SHA produit un résumé 160 bits, il est plus résistant aux attaques brutales que le hachage 128 bits (comme le MD5), mais il est plus lent.

Transmission tunnel partagée - Processus qui permet à un utilisateur d'un VPN à distance d'accéder à un réseau public, plus généralement Internet, tout en accédant à des ressources du bureau distant. Cette méthode d'accès réseau permet à l'utilisateur d'accéder à des équipements distants, tels qu'une imprimante reliée au réseau et des serveurs tout en accédant au réseau public (Internet). Un avantage de l'utilisation de la transmission tunnel partagée est l'allègement des goulots d'étranglement et le maintien de la bande passante car le trafic Internet ne doit pas passer par le serveur VPN. Un inconvénient de cette méthode est qu'il rend le VPN principalement vulnérable à l'attaque car il est accessible via le réseau public, non sécurisé.

Transform - Une transformation décrit un protocole de sécurité (AH ou ESP) avec ses algorithmes correspondants. Par exemple, ESP avec l'algorithme de chiffrement DES et HMAC-SHA pour l'authentification.

Transport Mode - Mode d'encapsulation pour AH/ESP. Transport Mode encapsule la charge utile supérieure de couche, telle que le Protocole de contrôle de transmissions (TCP) ou le Protocole de datagramme utilisateur (UDP), du datagramme IP original. Ce mode peut seulement être utilisé quand les homologues sont les points de terminaison de la communication. Le contraste de Transport Mode est Tunnel Mode.

Tunnel Mode - Encapsulation du datagramme IP complet pour IPsec. Tunnel Mode est utilisé pour protéger des datagrammes originaires ou destinés aux systèmes non IPsec, tels qu'un scénario de réseau privé virtuel (VPN).

Configurer ISAKMP

IKE existe seulement pour établir les SA pour IPsec. Avant de pouvoir faire cela, IKE doit négocier un rapport de SA (une SA ISAKMP) avec l'homologue. Puisqu'IKE négocie sa propre politique, il est possible de configurer des instructions de politique multiples avec différentes instructions de configuration, puis laisser les deux hôtes parvenir à un accord. ISAKMP négocie :

- **un algorithme de chiffrement** - Limité à DES 56 bits seulement ;
- **un algorithme de hachage** MD5 ou SHA ;
- Authentification - Signatures RSA, nombres aléatoires chiffrés RSA, ou clés pré-partagées ;
- **la durée de vie de SA** - En secondes.

Actuellement, il y a deux méthodes utilisées pour configurer ISAKMP :

1. Utiliser les clés pré-partagées, qui sont simples à configurer.
2. Utiliser une **CA**, qui est évolutive dans toute l'entreprise.

Remarque: La négociation IKE s'effectue sur l'UDP 500. IPsec utilise des protocoles IP 50 et 51. Assurez-vous que ceux-ci sont autorisés sur toutes les listes d'accès que vous avez entre les homologues.

1. Clés pré-partagées

C'est la méthode simple et rapide utilisée pour configurer IKE. Tandis que la configuration IKE est

simple et que vous n'utilisez pas une CA, l'évolution ne fonctionne pas très bien.

Vous devez effectuer ce qui suit afin de configurer IKE :

- Configurer la suite de protection ISAKMP.
- Configurer la clé ISAKMP.

Configurer la suite de protection ISAKMP

Cette commande crée l'objet de politique ISAKMP. Il est possible d'avoir plusieurs politiques, mais il y en a seulement une dans cet exemple :

```
dt3-45a(config)#crypto isakmp policy 1 dt3-45a(config-isakmp)#
```

Avec la commande **group**, vous pouvez déclarer quel module de taille à l'utiliser pour le calcul de Diffie-Hellman. La longueur du groupe 1 est 768 bits et celle du groupe 2 est 1024 bits. Pourquoi utiliseriez-vous l'un sur l'autre ? Tous les constructeurs ne prennent pas en charge le groupe 2. En outre, le groupe 2 est aussi manifestement plus intensif au niveau du CPU que le groupe un. Pour cette raison, vous ne voulez pas utiliser le groupe 2 sur des routeurs bas de gamme comme les routeurs Cisco 2500 ou inférieurs. Cependant, le groupe 2 est plus sécurisé que le groupe 1. Puisque cet exemple utilise un Cisco 4500, le groupe 2 est utilisé, et il convient de s'assurer que l'homologue est également configuré pour utiliser le groupe 2. Le groupe par défaut est le groupe 1. Si vous sélectionnez les propriétés par défaut, les lignes du groupe 1 n'apparaissent pas quand vous faites une commande **write terminal**.

```
dt3-45a(config-isakmp)#group 2
```

MD5 est notre algorithme de hachage dans cette ligne. Tandis que l'implémentation du SHA et MD5 sont obligatoires, tous les homologues ne peuvent pas être configurés afin de négocier l'un ou l'autre. SHA est la configuration par défaut dans Cisco IOS, qui est plus sécurisé que MD5.

```
dt3-45a(config-isakmp)#hash md5
```

La durée de vie de SA, 500 secondes dans ce cas, est affichée dans cette commande. Si vous ne configurez pas de durée de vie, la valeur par défaut est 86400 secondes, ou un jour. Quand le temporisateur de durée de vie se déclenche, SA est renégociée comme mesure de sécurité.

```
dt3-45a(config-isakmp)#lifetime 500
```

Dans cette commande, la clé à utiliser est indiquée manuellement à IKE. Par conséquent, la commande **pre-share** est utilisée. Deux options, sans compter la commande **pre-share**, sont les commandes **rsa-encr** et **rsa-sig**. La commande **rsa-encr** configure des nombres aléatoires chiffrés RSA et la commande **rsa-sig** configure la signature RSA. Les commandes **rsa-encr** et **rsa-sig** sont abordées dans la section Utiliser une autorité de certification (CA). Pour le moment, rappelez-vous que **rsa-sig** est la commande par défaut.

```
dt3-45a(config-isakmp)#authentication pre-share
```

Configurer la clé ISAKMP

Dans ces commandes, la clé à utiliser est indiquée à IKE. L'homologue, 192.168.10.38 dans ce cas, doit avoir la même clé Slurpee-Machine dans sa configuration.

```
dt3-45a(config-isakmp)#exit dt3-45a(config)#crypto isakmp key Slurpee-Machine address 192.168.10.38
```

Vous avez terminé la configuration IKE. Ces lignes sont la configuration IKE de l'homologue. Les configurations complètes pour les deux routeurs figurent dans la section Exemples de configuration de ce document :

```
crypto isakmp policy 1
 hash md5
 group 2
 authentication pre-share
crypto isakmp key Slurpee-Machine address 192.168.10.66
```

2. Utiliser une autorité de certification (CA)

L'utilisation d'une CA est une méthode complexe utilisée afin de configurer IKE. Puisqu'elle est très évolutive dans IPsec, vous devez utiliser IPsec au lieu du chiffrement classique. Quand le Logiciel Cisco IOS Version 11.3(3) sera publié, seuls quelques constructeurs de CA commercialiseront le produit. Au commencement, la plupart des configurations sont faites avec l'utilisation des **clés pré-partagées**. VeriSign, Entrust, Microsoft et Netscape, et probablement bien d'autres encore, fonctionnent sur des produits CA. Pour cet exemple, une CA VeriSign est utilisée.

Vous devez effectuer ce qui suit afin d'utiliser une CA :

- Créer une paire de clés RSA pour le routeur.
- Demander un certificat de CA.
- Inscrire les certificats pour le routeur client.
- Configurer la suite de protection ISAKMP.

Créer des paires de clés RSA pour le routeur

La commande **crypto key gen rsa usage-keys** peut vous troubler. Cette commande crée deux paires de clés pour RSA :

- une paire de clés pour le chiffrement
- une paire de clés pour des signatures numériques

Une paire de clés se rapporte à une clé publique et à sa clé secrète correspondante. Si vous ne spécifiez pas usage-keys à la fin de la commande, le routeur génère seulement une paire de clés RSA et l'utilise à la fois pour le chiffrement et les signatures numériques. Il faut savoir que cette commande peut être utilisée pour créer des clés DSS. Mais DSS est une partie du chiffrement classique, pas IPsec.

```
dt3-45a(config)#crypto key gen rsa usage-keys The name for the keys will be: dt3-45a.cisco.com %You
already have RSA keys defined for dt3-45a.cisco.com. %Do you really want to replace them? [yes/no] yes
```

Puisque les clés RSA existent déjà dans la zone, il demande si vous voulez vous débarrasser des clés existantes. Puisque la réponse est positive, confirmez la commande. Cette invite est retournée :

```
Choose the size of the key modulus in the range of
   360 to 2048 for your Signature keys.
Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: <return>
Generating RSA keys...
[OK]

Choose the size of the key modulus in the range of
   360 to 2048 for your Encryption keys.
Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: <return>
Generating RSA keys...
[OK]
```

```
dt3-45a(config)#
```

Les paires de clés RSA avec le module 512 bits par défaut sont maintenant créées. Quittez le mode de configuration et entrez une commande **show crypto key mypubkey rsa**. Vous pouvez maintenant voir votre clé publique RSA. La partie clé privée de la paire de clés n'est jamais vue. Même si vous n'avez pas de clés préexistantes, vous voyez la même chose que précédemment.

Remarque: Rappelez-vous de sauvegarder votre configuration une fois que vous avez produit votre paire de clés.

Demander un certificat de CA

Vous devez maintenant configurer le routeur afin de parler à une CA. Ceci implique plusieurs étapes. Vous devez vous coordonner par la suite avec votre administrateur de CA.

Dans ces lignes de configuration, un nom de domaine est ajouté au routeur. Un nom d'hôte **ciscoca-ultra** est créé, et le routeur prend connaissance de son adresse IP et des serveurs de noms. Vous devez avoir, soit des noms d'hôte définis pour la CA, soit un DNS qui fonctionne dans la zone. Cisco recommande que vous ayez un DNS qui fonctionne dans la zone.

```
dt3-45a(config)#ip host ciscoca-ultra 171.69.54.46 dt3-45a(config)#ip domain-name cisco.com dt3-45a(config)#ip name-server 171.692.132 dt3-45a(config)#ip name-server 198.92.30.32
```

Commencer à configurer les paramètres de CA. **verisign-ca** est juste un nom arbitraire.

```
dt3-45a(config)#crypto ca identity verisign-ca dt3-45a(ca-identity)#
```

Dans cette sortie, le protocole d'inscription Cisco utilise HTTP afin de parler à la CA. La commande **dt3-45a(ca-identity)#enrollment url http://ciscoca-ultra** indique au routeur d'aller à l'URL spécifique afin d'interagir avec la CA. La commande **dt3-45a(ca-identity)#crypto ca authenticate verisign-ca** indique au routeur de chercher le certificat de CA. Avant que vous puissiez s'inscrire dans le CA, vous devez vous veiller entretien au vrai CA pour vérifier le certificat du CA avec l'administrateur CA afin d'assurer l'authenticité.

```
dt3-45a(ca-identity)#enrollment url http://ciscoca-ultra dt3-45a(ca-identity)#exit dt3-45a(ca-identity)#crypto ca authenticate verisign-ca
```

Inscrire les certificats pour le routeur client

Émettez la commande **crypto ca enroll verisign-ca** afin de commencer l'inscription avec la CA. Il y a plusieurs étapes pour cela. D'abord, vous devez vérifier l'identité de la CA, puis la CA doit vérifier l'identité du routeur. Si vous devez annuler votre certificat avant qu'il expire, si vous renumérotez les interfaces du votre routeur ou si vous croyez que votre certificat est compromis, vous devez fournir un mot de passe à l'administrateur de CA. Entrez cela, comme cela est illustré dans cette sortie. Après que vous avez entré votre mot de passe, le routeur continue.

```
dt3-45a(config)#crypto ca enroll verisign-ca %Start certificate enrollment .. %Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a note of it. Password: Re-enter password:
```

Vous voyez maintenant les empreintes digital du CA vérifier que les empreintes digital sont correctes avec l'administrateur CA. En outre, si vous effectuez une commande **show crypto ca cert**, vous voyez le certificat de CA, en plus de vos propres certificats. Les certificats de CA sont alors mentionnés comme étant en attente.

```
% The subject name for the keys will be: dt3-45a.cisco.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: 01204044
```

```
% Include an IP address in the subject name? [yes/no]: yes
Interface: Ethernet 0
Request certificate from CA? [yes/no]: yes
```

Contactez l'administrateur de CA parce que cette personne veut confirmer l'identité du tuyau avant qu'un certificat soit émis. Une fois que la CA a émis le certificat, l'état en attente de notre certificat change et devient disponible. Cela conclut l'inscription de CA. Mais, vous n'avez pas terminé. Vous devez toujours configurer l'objet de politique ISAKMP.

Configurer la suite de protection ISAKMP

La commande **Rsa-sig** par défaut est utilisée dans cette sortie. Vous pouvez avoir plusieurs suites de protection, mais il y en a seulement une dans cet exemple. Dans le cas de suites de protection multiples, les politiques sont présentées à l'homologue dans l'ordre numérique et l'homologue négocie celui à utiliser. Vous devez faire cela si vous savez que tous vos homologues ne prennent pas en charge certaines fonctionnalités. Le routeur n'essaye pas de négocier ce qui semble déraisonnable. Par exemple, si vous configurez votre politique pour **rsa-sig** et que vous n'avez aucun certificat, le routeur ne négocie pas cela.

```
dt3-45a(config)#crypto isakmp policy 1 dt3-45a(config-isakmp)#hash md5 dt3-45a(config-isakmp)#lifetime 4000 dt3-45a(config-isakmp)#exit
```

Configurer IPsec

Si vous utilisez des clés pré-partagées ou configurez une CA, une fois que vous configurez l'Échange de clés Internet (IKE), vous devez encore configurer IPsec. Indépendamment de la méthode IKE utilisée, les étapes de configuration pour IPsec sont les mêmes.

Vous devez effectuer ce qui suit afin de configurer IPsec :

- Créer une liste de contrôle d'accès étendue (ACL).
- Créer une transformation IPsec.
- Créer une carte de chiffrement.
- Appliquer une carte de chiffrement à l'interface.

Créer une liste de contrôle d'accès étendue (ACL)

Cette commande est une ACL très simple qui permet aux routeurs de dialoguer, par exemple, le Telnet d'un routeur au suivant.

```
dt3-45a(config)#access-list 101 permit ip host 192.168.10.38 host 192.168.10.66
```

Une ACL plus réaliste ressemble à cette commande. Cette commande est une liste de contrôle d'accès étendue ordinaire, où 192.168.3.0 est un sous-réseau derrière le routeur en question, et 10.3.2.0 est un sous-réseau quelque part derrière le routeur homologue. Rappelez-vous que **permit** signifie chiffrer et **deny** signifie ne pas chiffrer.

```
dt3-45a(config)#access-list 101 permit ip 192.168.3.0 0.0.0.255 10.3.2.0 0.0.0.255
```

Créer une transformation IPsec

Créer trois jeux de transformation. Le premier utilise ESP seulement, le deuxième utilise AH combiné avec ESP et le dernier utilise seulement AH. Lors de la négociation SA IPsec, chacun

des trois est offert à l'homologue, qui en choisit un. En outre, pour chacun des jeux de transformation, utilisez le **tunnel mode** par défaut. Le mode de transport peut être utilisé seulement quand les points de terminaison de chiffrement sont également les points de terminaison de la communication. Le mode de transport peut être spécifié par la commande **mode transport** sous la configuration du jeu de transformation. Le mode tunnel est utilisé principalement pour le scénario VPN. Notez également que **esp-rfc1829** et **ah-rfc1828** sont basés sur le RFC initial pour cette technologie et sont des transformations obsolètes incluses pour la rétrocompatibilité. Tous les constructeurs ne prennent pas en charge ces dernières transformations, mais d'autres constructeurs ne prennent en charge que ces transformations.

Les jeux de transformation dans ces commandes ne sont pas nécessairement les plus pratiques. Par exemple, PapaBear et BabyBear ont des jeux de transformation non conformes aux normes. Utilisez **esp-rfc1829** et **ah-rfc1828** ensemble dans le même jeu de transformation.

```
dt3-45a(config)#crypto ipsec transform-set PapaBear esp-rfc1829 dt3-45a(cfg-crypto-trans)#exit dt3-
45a(config)#crypto ipsec transform-set MamaBear ah-md5-hmac esp-des dt3-45a(cfg-crypto-trans)#exit dt3-
45a(config)#crypto ipsec transform-set BabyBear ah-rfc1828 dt3-45a(cfg-crypto-trans)#exit dt3-
45a(config)#
```

Créer une carte de chiffrement

La balise **ipsec-isakmp** indique au routeur que cette carte de chiffrement est une carte de chiffrement IPsec. Bien qu'il y ait seulement un homologue déclaré dans cette carte de chiffrement, vous pouvez avoir plusieurs homologues dans une carte de chiffrement donnée. La **session key lifetime** peut être exprimée en kilo-octets (après x-quantité de trafic, changez la clé) ou en secondes, comme cela est affiché dans ces commandes. L'objectif est de rendre les efforts d'un attaquant potentiel plus difficiles. La commande **set transform-set** sert à associer les transformations avec la carte de chiffrement. En outre, l'ordre dans lequel vous déclarez les transformations est significatif. MamaBear est préférée dans cette configuration, puis le reste dans l'ordre de préférence décroissant jusqu'à BabyBear. La commande **match address 101** permet d'utiliser la liste d'accès 101 afin de déterminer le trafic qui est pertinent. Vous pouvez avoir plusieurs cartes de chiffrement portant le même nom, armadillo, dans cet exemple et différents numéros de séquence, 10, dans cet exemple. La combinaison de plusieurs cartes de chiffrement et des différents numéros de séquence vous permet de mélanger et de faire correspondre le chiffrement classique et IPsec. Vous pouvez également modifier votre configuration PFS ici. Le group1 PFS est la valeur par défaut dans cet exemple. Vous pouvez changer le PFS en group2, ou le désactiver complètement, ce que vous ne devriez pas faire.

```
dt3-45a(config)#crypto map armadillo 10 ipsec-isakmp dt3-45a(config-crypto-map)#set peer 192.168.10.38
dt3-45a(config-crypto-map)#set session-key lifetime seconds 4000 dt3-45a(config-crypto-map)#set
transform-set MamaBear PapaBear BabyBear dt3-45a(config-crypto-map)#match address 101
```

Appliquer une carte de chiffrement à l'interface

Ces commandes appliquent la carte de chiffrement à l'interface. Vous pouvez assigner seulement une carte de chiffrement à une interface. Si plusieurs entrées de carte de chiffrement ont le même nom de carte mais un numéro de séquence différent, elles font partie du même ensemble et s'appliquent toutes à l'interface. L'appliance de sécurité évalue l'entrée de **crypto map** avec le numéro de séquence le plus bas en premier.

```
dt3-45a(config)#interface e0 dt3-45a(config-if)#crypto map armadillo
```

Considérations sur la mémoire et le CPU

Les paquets qui sont traités par IPsec sont plus lents que les paquets traités par chiffrement

classique. Il y a plusieurs raisons à cela et ils pourraient poser des problèmes de performances significatifs :

1. IPsec introduit l'extension de paquet, qui est davantage susceptible d'exiger la fragmentation et le réassemblage correspondant des datagrammes IPsec.
2. Les paquets chiffrés sont probablement authentifiés, ce qui signifie qu'il y a deux opérations cryptographiques qui sont exécutées pour chaque paquet.
3. Les algorithmes d'authentification sont lents, bien que le travail ait été effectué pour accélérer des choses comme les calculs de Diffie-Hellman.

En outre, l'échange de clés de Diffie-Hellman utilisé dans l'IKE est une croissance exponentielle des nombres très grands (entre 768 et 1024 octets) et peut prendre jusqu'à quatre secondes sur Cisco 2500. La performance de RSA dépend de la taille du nombre premier choisi pour les paires de clés RSA.

Pour chaque routeur, la base de données SA prend approximativement 300 octets, plus 120 octets pour chaque SA suivante. Dans les situations où il existe deux SA IPsec, une entrante et une sortante, 540 octets sont requis, dans la plupart des cas. Chaque entrée SA IKE correspond approximativement à 64 octets. Le seul cas où vous avez un SA IPsec pour un flux de données est quand la communication est à sens unique.

IPsec et IKE ont des conséquences sur la performance lors de l'activité. Les échanges de clés de Diffie-Hellman, l'authentification de clé publique, et le chiffrement/déchiffrement consomment une importante quantité de ressources. Pourtant de nombreux efforts ont été accomplis pour limiter cette incidence.

Il y a une petite diminution de performance pour des paquets non chiffrés qui passent par une interface qui chiffre. C'est parce que tous les paquets doivent être contrôlés par rapport à la carte de chiffrement. Il n'y a aucune incidence de performance pour les paquets qui traversent le routeur qui évite une interface qui chiffre. L'impact important est sur les flux de données chiffrées.

Employez Group 1 pour des échanges de clés Diffie-Hellman dans IKE, utilisez MD5 en tant qu'algorithme de hachage et utilisez des durées de vie plus longues afin de limiter l'incidence du sous-système de chiffrement sur le reste du routeur. Pour équilibrer ce paramétrage de performances, vous pouvez baisser le niveau de chiffrement. Finalement, c'est la politique de sécurité du client qui détermine les fonctionnalités à utiliser et celles à mettre de côté.

Sortie de commandes show

Remarque: Les saisies dans ces sections sont tirées d'une série de tests différents que ceux utilisés dans les sections précédentes de ce document. En conséquence, ces saisies peuvent avoir différentes adresse IP et refléter des configurations légèrement différentes. D'autres séries de commandes **show** sont fournies dans la section Informations de débogage de ce document.

Sortie liée à IKE

Étudiez ces commandes afin de contrôler l'inscription de la CA Verisign. Ces commandes montrent les clés publiques que vous utilisez pour le chiffrement et les signatures RSA.

```
dt1-45a#show crypto key mypubkey rsa % Key pair was generated at: 11:31:59 PDT Apr 9 1998 Key name: dt1-
45a.cisco.com Usage: Signature Key Key Data: 305C300D 06092A86 4886F70D 01010105 00034B00 30480241
00C11854 39A9C75C 4E34C987 B4D7F36C A058D697 13172767 192166E1 661483DD 0FDB907B F9C10B7A CB5A034F
```

```
A41DF385 23BEB6A7 C14344BE E6915A12 1C86374F 83020301 0001 % Key pair was generated at: 11:32:02 PDT Apr 9 1998 Key name: dt1-45a.cisco.com Usage: Encryption Key Key Data: 305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DCF5AC 360DD5A6 C69704CF 47B2362D 65123BD4 424B6FF6 AD10C33E 89983D08 16F1EA58 3700BCF9 1EF17E71 5931A9FC 18D60D9A E0852DDD 3F25369C F09DFB75 05020301 0001
```

Cette commande montre les certificats que le routeur reconnaît. Un certificat dont l'état est **Pending** a été soumis à la CA pour approbation.

```
dt1-45a#show crypto ca certificates Certificate Subject Name Name: dt1-45a.cisco.com Serial Number: 01193485 Status: Available Certificate Serial Number: 650534996414E2BE701F4EF3170EDFAD Key Usage: Signature CA Certificate Status: Available Certificate Serial Number: 3051DF7169BEE31B821DFE4B3A338E5F Key Usage: Not Set Certificate Subject Name Name: dt1-45a.cisco.com Serial Number: 01193485 Status: Available Certificate Serial Number: 1e621faf3b9902bc5b49d0f99dc66d14 Key Usage: Encryption
```

Cette sortie montre les clés publiques du routeur et où ce dernier en a pris connaissance.

```
dt1-45a#show crypto key pubkey-chain rsa Codes: M - Manually configured, C - Extracted from certificate Code Usage IP-Address Name C Signing Cisco SystemsDevtestCISCOCA-ULTRA C General 172.21.30.71 dt1-7ka.cisco.com
```

C'est la table SA (IKE) ISAKMP. Vous voyez qu'une SA existe actuellement entre 172.21.30.71 et 172.21.30.70. L'homologue doit avoir une entrée de SA dans le même état que la sortie de ce routeur.

```
dt1-7ka#show crypto isakmp sa dst src state conn-id slot 172.21.30.70 172.21.30.71 QM_IDLE 47 5
```

Ces lignes montrent les objets de politique configurés. Dans ce cas, les politiques **1**, **2** et **4** sont utilisées, en plus des politiques par défaut. Les politiques sont proposées à l'homologue dans l'ordre, **1** étant la préférée.

```
dt1-45a#show crypto isakmp policy Protection suite of priority 1 encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Message Digest 5 authentication method: Rivest-Shamir-Adleman Signature Diffie-Hellman group: #1 (768 bit) lifetime: 180 seconds, no volume limit Protection suite of priority 2 encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Secure Hash Standard authentication method: Pre-Shared Key Diffie-Hellman group: #2 (1024 bit) lifetime: 180 seconds, no volume limit Protection suite of priority 4 encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Message Digest 5 authentication method: Pre-Shared Key Diffie-Hellman group: #2 (1024 bit) lifetime: 180 seconds, no volume limit Default protection suite encryption algorithm: DES - Data Encryption Standard (56 bit keys). hash algorithm: Secure Hash Standard authentication method: Rivest-Shamir-Adleman Signature Diffie-Hellman group: #1 (768 bit) lifetime: 86400 seconds, no volume limit
```

Commandes show liées à IPsec

Cette commande montre la carte de chiffrement **ToOtherRouter**, les ACL et les propositions de transformation appliquées à cette carte de chiffrement, aux homologues et à la durée de vie de la clé.

```
S3-2513-2#show crypto map Crypto Map "ToOtherRouter" 10 ipsec-isakmp Peer = 192.168.1.1 Extended IP access list 101 access-list 101 permit ip source: addr = 192.168.45.0/0.0.0.255 dest: addr = 192.168.3.0/0.0.0.255 Connection Id = UNSET (0 established, 0 failed) Current peer: 192.168.1.1 Session key lifetime: 4608000 kilobytes/3600 seconds PFS (Y/N): N Transform proposals={ Elvis, Bubba, BarneyDino, }
```

Cette configuration utilise le même routeur que la sortie précédente, mais des commandes différentes. Vous voyez toutes les propositions de transformation, les paramètres qu'elles négocient et les valeurs par défaut.

```
S3-2513-2#show crypto ipsec transform-set Transform proposal Elvis: { ah-sha-hmac } supported settings = { Tunnel, }, default settings = { Tunnel, }, will negotiate = { Tunnel, }, { esp-des } supported settings = { Tunnel, }, default settings = { Tunnel, }, will negotiate = { Tunnel, }, Transform proposal Bubba: { ah-rfc1828 } supported settings = { Tunnel, }, default settings = { Tunnel, }, will negotiate = { Tunnel, }, { esp-des esp-md5-hmac } supported settings = { Tunnel, }, default settings = { Tunnel, }, will negotiate = { Tunnel, }, Transform proposal BarneyDino: { ah-md5-hmac } supported settings = { Tunnel, },
```

```
default settings = { Tunnel, }, will negotiate = { Tunnel, },
```

Cette commande affiche les associations de sécurité actuelles IPsec de ce routeur. Un routeur comporte une SA AH pour les flux entrant et sortant.

```
S3-2513-2#show crypto ip session Session key lifetime: 4608000 kilobytes/3600 seconds S3-2513-2#show
crypto ipsec sa interface: Ethernet0 Crypto map tag: ToOtherRouter, local addr. 192.168.1.2 local ident
(addr/mask/prot/port): (192.168.45.0/255.255.255.0/0/0) remote ident (addr/mask/prot/port):
(192.168.3.0/255.255.255.0/0/0) current_peer: 192.168.1.1 PERMIT, flags={origin_is_acl,} #pkts encaps: 0,
#pkts encrypt: 0, #pkts digest 0 #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0 #send errors 5, #recv
errors 0 local crypto endpt.: 192.168.1.2, remote crypto endpt.: 192.168.1.1 path mtu 1500, media mtu
1500 current outbound spi: 25081A81 inbound esp sas: inbound ah sas: spi: 0x1EE91DDC(518594012)
transform: ah-md5-hmac , in use settings ={Tunnel, } slot: 0, conn id: 16, crypto map: ToOtherRouter sa
timing: remaining key lifetime (k/sec): (4608000/3423) replay detection support: Y outbound esp sas:
outbound ah sas: spi: 0x25081A81(621288065) transform: ah-md5-hmac , in use settings ={Tunnel, } slot: 0,
conn id: 17, crypto map: ToOtherRouter sa timing: remaining key lifetime (k/sec): (4608000/3424) replay
detection support: Y
```

Exemples de configuration

Cette configuration utilise des **clés pré-partagées**. Cette configuration du routeur est utilisée afin de créer la sortie de débogage mentionnée dans la section Informations de débogage. Cette configuration permet à un réseau appelé X situé derrière le routeur d'origine de parler à un réseau appelé Y situé derrière le routeur homologue. Consultez la documentation du Logiciel Cisco IOS pour connaître la version de votre Cisco IOS, ou utilisez l'Outil de recherche de commande (clients enregistrés seulement) pour plus d'informations sur une commande spécifique. Cet outil permet à l'utilisateur de rechercher une description détaillée ou des directives de configuration pour une commande spécifique.

Diagramme du réseau

Configurations

- Routeur d'origine
- Routeur homologue

Routeur d'origine

```
Current configuration:
!
version 11.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname goss-e4-2513
!
enable secret 5 $1$ZuRD$YBaAh3oIv4iltIn0TMCUX1
enable password ww
!
!--- IKE configuration crypto isakmp policy 1 hash md5
authentication pre-share crypto isakmp key Slurpee-Machine
address 20.20.20.21 ! !--- IPsec configuration crypto ipsec
transform-set BearPapa esp-rfc1829 crypto ipsec transform-set
BearMama ah-md5-hmac esp-des crypto ipsec transform-set
BearBaby ah-rfc1828 ! crypto map armadillo 1 ipsec-isakmp set
peer 20.20.20.21 set security-association lifetime seconds
190 set transform-set BearPapa BearMama BearBaby !--- Traffic
to encrypt match address 101 ! interface Ethernet0 ip address
```

```
60.60.60.60 255.255.255.0 no mop enabled ! interface Serial0
ip address 20.20.20.20 255.255.255.0 no ip mroute-cache no
fair-queue crypto map armadillo ! interface Serial1 no ip
address shutdown ! interface TokenRing0 no ip address
shutdown ! ip classless ip route 0.0.0.0 0.0.0.0 20.20.20.21
!--- Traffic to encrypt access-list 101 permit ip 60.60.60.0
0.0.0.255 50.50.50.0 0.0.0.255 dialer-list 1 protocol ip
permit dialer-list 1 protocol ipx permit ! line con 0 exec-
timeout 0 0 line aux 0 line vty 0 4 password ww login ! end
```

Routeur homologue

```
Current configuration:
!
version 11.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname goss-c2-2513
!
enable secret 5 $1$DBTl$Wtg2eS7Eb/Cw5l.nDhkEi/
enable password ww
!
ip subnet-zero
!
!--- IKE configuration crypto isakmp policy 1 hash md5
authentication pre-share crypto isakmp key Slurpee-Machine
address 20.20.20.20 ! !--- IPsec configuration crypto ipsec
transform-set PapaBear esp-rfc1829 crypto ipsec transform-set
MamaBear ah-md5-hmac esp-des crypto ipsec transform-set
BabyBear ah-rfc1828 ! ! crypto map armadillo 1 ipsec-isakmp
set peer 20.20.20.20 set security-association lifetime
seconds 190 set transform-set MamaBear PapaBear BabyBear !---
Traffic to encrypt match address 101 ! ! ! interface
Ethernet0 ip address 50.50.50.50 255.255.255.0 no ip
directed-broadcast ! interface Serial0 ip address 20.20.20.21
255.255.255.0 no ip directed-broadcast no ip mroute-cache no
fair-queue clockrate 9600 crypto map armadillo ! interface
Serial1 no ip address no ip directed-broadcast shutdown !
interface TokenRing0 no ip address no ip directed-broadcast
shutdown ! ip classless ip route 0.0.0.0 0.0.0.0 20.20.20.20
!--- Traffic to encrypt access-list 101 permit ip 50.50.50.0
0.0.0.255 60.60.60.0 0.0.0.255 dialer-list 1 protocol ip
permit dialer-list 1 protocol ipx permit ! ! line con 0 exec-
timeout 0 0 transport input none line aux 0 line aux 0 line
vty 0 4 password ww login ! end
```

Informations de débogage

Cette section comporte la sortie de débogage d'une session IKE/IPsec normale entre deux routeurs. Les configurations proviennent de la section Exemples de configuration de ce document. Les routeurs utilisent une clé pré-partagée. Les commandes **debug crypto isakmp**, **debug crypto ipsec** et **debug crypto engine** des deux routeurs sont activées. Cela a été testé avec un ping étendu de l'interface Ethernet du routeur d'origine vers l'interface Ethernet du routeur homologue (60.60.60.60 à 50.50.50.50).

Remarque: Les déclarations en bleu et en italiques dans cet exemple de sortie de débogage sont des notes permettant de vous aider à suivre ce qui se produit, elles ne font pas partie de la sortie de débogage.

- Routeur d'origine
- Sortie de la commande show du routeur d'origine après négociation IKE/IPsec
- Routeur homologue avec la même séquence de ping, comme vu de l'autre côté
- Commandes **show du** routeur homologue

Routeur d'origine

goss-e4-2513#**show clock** goss-e4-2513#**ping** Protocol [ip]: Target IP address: **50.50.50.50** Repeat count [5]: **10** Datagram size [100]: Timeout in seconds [2]: Extended commands [n]: **y** Source address or interface: **60.60.60.60** 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 10, 100-byte ICMP Echos to 50.50.50.50, timeout is 2 seconds: Apr 2 12:03:55.347: IPSEC(sa_request): , (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21, src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004 Apr 2 12:03:55.355: IPSEC(sa_request): , (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21, src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= AH, transform= ah-md5-hmac , lifedur= 190s and 4608000kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004 Apr 2 12:03:55.363: IPSEC(sa_request): , (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21, src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-des , lifedur= 190s and 4608000kb, spi= 0x0(0), conn_id= 0, keysi.ze= 0, flags= 0x4004 Apr 2 12:03:55.375: IPSEC(sa_request): , (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21, src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= AH, transform= ah-rfc1828 , lifedur= 190s and 4608000kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004 *!--- Note that the router offers to the peer all of the !--- available transforms.* Apr 2 12:03:55.391: ISAKMP (14): beginning Main Mode exchange Apr 2 12:03:57.199: ISAKMP (14): processing SA payload. message ID = 0 Apr 2 12:03:57.203: ISAKMP (14): Checking ISAKMP transform 1 against priority 1 policy Apr 2 12:03:57.203: ISAKMP: encryption DES-CBC Apr 2 12:03:57.207: ISAKMP: hash MD5 Apr 2 12:03:57.207: ISAKMP: default group 1 Apr 2 12:03:57.207: ISAKMP: auth pre-share Apr 2 12:03:57.211: ISAKMP (14): atts are acceptable. Next payload is 0 Apr 2 12:03:57.215: Crypto engine 0: generate alg param Apr 2 12:03:5.8.867: CRYPTO ENGINE: Dh phase 1 status: 0 Apr 2 12:03:58.871: ISAKMP (14): SA is doing pre-shared key authentication.. Apr 2 12:04:01.291: ISAKMP (14): processing KE payload. message ID = 0 Apr 2 12:04:01.295: Crypto engine 0: generate alg param Apr 2 12:04:03.343: ISAKMP (14): processing NONCE payload. message ID = 0 Apr 2 12:04:03.347: Crypto engine 0: create ISAKMP SKEYID for conn id 14 Apr 2 12:04:03.363: ISAKMP (14): SKEYID state generated Apr 2 12:04:03.367: ISAKMP (14): processing vendor id payload Apr 2 12:04:03.371: ISAKMP (14): speaking to another IOS box! Apr 2 12:04:03.371: generate hmac context for conn id 14 Apr 2 12:04:03.615: ISAKMP (14): processing ID payload. message ID = 0 Apr 2 12:04:03.615: ISAKMP (14): processing HASH payload. message ID = 0 Apr 2 12:04:03.619: generate hmac context for

conn id 14 Apr 2 12:04:03.627: ISAKMP (14): SA has been authenticated Apr 2 12:04:03.627: ISAKMP (14): beginning Quick Mode exchange, M-ID of 1628162439 *!--- These lines represent verification that the policy !--- attributes are fine, and the final authentication of the IKE SA. !--- Once the IKE SA is authenticated, a valid IKE SA exists. !--- New IKE kicks off IPsec negotiation:* Apr 2 12:04:03.635: IPSEC(key engine): got a queue event... Apr 2 12:04:03.635: IPSEC(spi response): getting spi 303564824ld for SA .!!!from 20.20.20.21 to 20.20.20.20 for prot 3 Apr 2 12:04:03.639: IPSEC(spi response): getting spi 423956280ld for SA from 20.20.20.21 to 20.20.20.20 for prot 2 Apr 2 12:04:03.643: IPSEC(spi response): getting spi 415305621ld for SA from 20.20.20.21 to 20.20.20.20 for prot 3 Apr 2 12:04:03.647: IPSEC(spi response): getting spi 218308976ld for SA from 20.20.20.21 to 20.20.20.20 for prot 2 Apr 2 12:04:03.891: generate hmac context for conn id 14 Apr 2 12:04:04.!! Success rate is 50 percent (5/10), round-trip min/avg/max = 264/265/268 ms goss-e4-2513#723: generate hmac context for conn id 14 Apr 2 12:04:04.731: ISAKMP (14): processing SA payload. message ID = 1628162439 Apr 2 12:04:04.731: ISAKMP (14): Checking IPSec proposal 1 Apr 2 12:04:04.735: ISAKMP: transform 1, ESP_DES_IV64 Apr 2 12:04:04.735: ISAKMP: attributes in transform: Apr 2 12:04:04.735: ISAKMP: encaps is 1 Apr 2 12:04:04.739: ISAKMP: SA life type in seconds Apr 2 12:04:04.739: ISAKMP: SA life duration (basic) of 190 Apr 2 12:04:04.739: ISAKMP: SA life type in kilobytes Apr 2 12:04:04.743: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 Apr 2 12:04:04.747: ISAKMP (14): atts are acceptable. *!--- The ISAKMP debug is listed because IKE is the !--- entity that negotiates IPsec SAs on behalf of IPsec.* Apr 2 12:04:04.747: IPSEC(validate_proposal_request): proposal part #1, (key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20, dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 0s and 0kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4 Apr 2 12:04:04.759: ISAKMP (14): processing NONCE payload. message ID = 1628162439 Apr 2 12:04:04.759: ISAKMP (14): processing ID payload. message ID = 1628162439 Apr 2 12:04:04.763: ISAKMP (14): processing ID payload. message ID = 1628162439 Apr 2 12:04:04.767: generate hmac context for conn id 14 Apr 2 12:04:04.799: ISAKMP (14): Creating IPSec SAs Apr 2 12:04:04.803: inbound SA from 20.20.20.21 to 20.20.20.20 (proxy 50.50.50.0 to 60.60.60.0) Apr 2 12:04:04.803: has spi 303564824 and conn_id 15 and flags 4 Apr 2 12:04:04.807: lifetime of 190 seconds Apr 2 12:04:04.807: lifetime of 4608000 kilobytes Apr 2 12:04:04.811: outbound SA from 20.20.20.20 to 20.20.20.21 (proxy 60.60.60.0 to 50.50.50.0) Apr 2 12:04:04.811: has spi 183903875 and conn_id 16 and flags 4 Apr 2 12:04:04.815: lifetime of 190 seconds Apr 2 12:04:04.815: lifetime of 4608000 kilobytes Apr 2 12:04:04.823: IPSEC(key_engine): got a queue event... Apr 2 12:04:04.823: IPSEC(initialize_sas): , (key eng. msg.) dest= 20.20.20.20, src= 20.20.20.21, dest_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), src_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi= 0x12180818(303564824), conn_id= 15, keysize= 0, flags= 0x4 Apr 2 12:04:04.831: IPSEC(initialize_sas): , (key eng. msg.) src= 20.20.20.20, dest= 20.20.20.21, src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi= 0xAF62683(183903875), conn_id= 16, keysize= 0, flags= 0x4 Apr

```
2 12:04:04.839: IPSEC(create_sa): sa created, (sa) sa_dest=
20.20.20.20, sa_prot= 50, sa_spi= 0x12180818(303564824),
sa_trans= esp-rfc1829 , sa_conn_id= 15 Apr 2 12:04:04.843:
IPSEC(create_sa): sa created, (sa) sa_dest= 20.20.20.21,
sa_prot= 50, sa_spi= 0xAF62683(183903875), sa_trans= esp-
rfc1829 , sa_conn_id= 16 !--- These lines show that IPsec SAs
are created and !--- encrypted traffic can now pass.
```

Sortie de la commande show du routeur d'origine après négociation IKE/IPsec

```
goss-e4-2513#
goss-e4-2513#show crypto isakmp sa dst src state conn-id slot
20.20.20.21 20.20.20.20 QM_IDLE 14 0 goss-e4-2513#show crypto
ipsec sa interface: Serial0 Crypto map tag: armadillo, local
addr. 20.20.20.20 local ident (addr/mask/prot/port):
(60.60.60.0/255.255.255.0/0/0) remote ident
(addr/mask/prot/port): (50.50.50.0/255.255.255.0/0/0)
current_peer: 20.20.20.21 PERMIT, flags={origin_is_acl,}
#pkts encaps: 5, #pkts encrypt: 5, #pkts digest 0 #pkts
decaps: 5, #pkts decrypt: 5, #pkts verify 0 #send errors 5,
#recv errors 0 local crypto endpt.: 20.20.20.20, remote
crypto endpt.: 20.20.20.21 path mtu 1500, media mtu 1500
current outbound spi: AF62683 inbound esp sas: spi:
0x12180818(303564824) transform: esp-rfc1829 , in use
settings ={Var len IV, Tunnel, } slot: 0, conn id: 15, crypto
map: armadillo sa timing: remaining key lifetime (k/sec):
(4607999/135) IV size: 8 bytes replay detection support: N
inbound ah sas: outbound esp sas: spi: 0xAF62683(183903875)
transform: esp-rfc1829 , in use settings ={Var len IV,
Tunnel, } slot: 0, conn id: 16, crypto map: armadillo sa
timing: remaining key lifetime (k/sec): (4607999/117) IV
size: 8 bytes replay detection support: N outbound ah sas:
goss-e4-2513#show crypto isakmp policy Protection suite of
priority 1 encryption algorithm: DES - Data Encryption
Standard (56 bit keys). hash algorithm: Message Digest 5
authentication method: Pre-Shared Key Diffie-Hellman group:
#1 (768 bit) lifetime: 86400 seconds, no volume limit Default
protection suite encryption algorithm: DES - Data Encryption
Standard (56 bit keys). hash algorithm: Secure Hash Standard
authentication method: Rivest-Shamir-Adleman Signature
Diffie-Hellman group: #1 (768 bit) lifetime: 86400 seconds,
no volume limit goss-e4-2513#show crypto map Crypto Map
"armadillo" 1 ipsec-isakmp Peer = 20.20.20.21 Extended IP
access list 101 access-list 101 permit ip 60.60.60.0
0.0.0.255 50.50.50.0 0.0.0.255 Current peer: 20.20.20.21
Security association lifetime: 4608000 kilobytes/190 seconds
PFS (Y/N): N Transform sets={ BearPapa, BearMama, BearBaby, }
```

Routeur homologue avec la même séquence de ping, comme vu de l'autre côté

```
goss-c2-2513#show debug Cryptographic Subsystem: Crypto
ISAKMP debugging is on Crypto Engine debugging is on Crypto
IPSEC debugging is on goss-c2-2513# Apr 2 12:03:55.107:
ISAKMP (14): processing SA payload. message ID = 0 Apr 2
12:03:55.111: ISAKMP (14): Checking ISAKMP transform 1
against priority 1 policy Apr 2 12:03:55.111: ISAKMP:
encryption DES-CBC Apr 2 12:03:55.111: ISAKMP: hash MD5 Apr 2
12:03:55.115: ISAKMP: default group 1 Apr 2 12:03:55.115:
ISAKMP: auth pre-share Apr 2 12:03:55.115: ISAKMP (14): atts
are acceptable. Next payload is 0 !--- IKE performs its
operation, and then kicks off IPsec. Apr 2 12:03:55.119:
Crypto engine 0: generate alg param Apr 2 12:03:56.707:
CRYPTO_ENGINE: Dh phase 1 status: 0 Apr 2 12:03:56.711:
ISAKMP (14): SA is doing pre-shared key authentication Apr 2
```

12:03:58.667: ISAKMP (14): processing KE payload. message ID = 0 Apr 2 12:03:58.671: Crypto engine 0: generate alg param Apr 2 12:04:00.687: ISAKMP (14): processing NONCE payload. message ID = 0 Apr 2 12:04:00.695: Crypto engine 0: create ISAKMP SKEYID for conn id 14 Apr 2 12:04:00.707: ISAKMP (14): SKEYID state generated Apr 2 12:04:00.711: ISAKMP (14): processing vendor id payload Apr 2 12:04:00.715: ISAKMP (14): speaking to another IOS box! Apr 2 12:04:03.095: ISAKMP (14): processing ID payload. message ID = 0 Apr 2 12:04:03.095: ISAKMP (14): processing HASH payload. message ID = 0 Apr 2 12:04:03.099: generate hmac context for conn id 14 Apr 2 12:04:03.107: ISAKMP (14): SA has been authenticated Apr 2 12:04:03.111: generate hmac context for conn id 14 Apr 2 12:04:03.835: generate hmac context for conn id 14 Apr 2 12:04:03.839: ISAKMP (14): processing SA payload. message ID = 1628162439 Apr 2 12:04:03.843: ISAKMP (14): Checking IPSec proposal 1 Apr 2 12:04:03.843: ISAKMP: transform 1, ESP_DES_IV64 Apr 2 12:04:03.847: ISAKMP: attributes in transform: Apr 2 12:04:03.847: ISAKMP: encaps is 1 Apr 2 12:04:03.847: ISAKMP: SA life type in seconds Apr 2 12:04:03.851: ISAKMP: SA life duration (basic) of 190 Apr 2 12:04:03.851: ISAKMP: SA life type in kilobytes Apr 2 12:04:03.855: ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0 Apr 2 12:04:03.855: ISAKMP (14): atts are acceptable. Apr 2 12:04:03.859: IPSEC(validate_proposal_request): proposal part #1, (key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20, dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 0s and 0kb, spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4 Apr 2 12:04:03.867: ISAKMP (14): processing NONCE payload. message ID = 1628162439 Apr 2 12:04:03.871: ISAKMP (14): processing ID payload. message ID = 1628162439 Apr 2 12:04:03.871: ISAKMP (14): processing ID payload. message ID = 1628162439 Apr 2 12:04:03.879: IPSEC(key_engine): got a queue event... Apr 2 12:04:03.879: IPSEC(spi_response): getting spi 183903875ld for SA from 20.20.20.20 to 20.20.20.21 for prot 3 Apr 2 12:04:04.131: generate hmac context for conn id 14 Apr 2 12:04:04.547: generate hmac context for conn id 14 Apr 2 12:04:04.579: ISAKMP (14): Creating IPSec SAs Apr 2 12:04:04.579: inbound SA from 20.20.20.20 to 20.20.20.21 (proxy 60.60.60.0 to 50.50.50.0) Apr 2 12:04:04.583: has spi 183903875 and conn_id 15 and flags 4 Apr 2 12:04:04.583: lifetime of 190 seconds Apr 2 12:04:04.587: lifetime of 4608000 kilobytes Apr 2 12:04:04.587: outbound SA from 20.20.20.21 to 20.20.20.20 (proxy 50.50.50.0 to 60.60.60.0) Apr 2 12:04:04.591: has spi 303564824 and conn_id 16 and flags 4 Apr 2 12:04:04.591: lifetime of 190 seconds Apr 2 12:04:04.595: lifetime of 4608000 kilobytes Apr 2 12:04:04.599: IPSEC(key_engine): got a queue event... Apr 2 12:04:04.599: IPSEC(initialize_sas): , (key eng. msg.) dest= 20.20.20.21, src= 20.20.20.20, dest_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), src_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi= 0xAF62683(183903875), conn_id= 15, keysize= 0, flags= 0x4 Apr 2 12:04:04.607: IPSEC(initialize_sas): , (key eng. msg.) src= 20.20.20.21, dest= 20.20.20.20, src_proxy= 50.50.50.0/255.255.255.0/0/0 (type=4), dest_proxy= 60.60.60.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform= esp-rfc1829 , lifedur= 190s and 4608000kb, spi= 0x12180818(303564824), conn_id= 16, keysize= 0, flags= 0x4 Apr 2 12:04:04.615: IPSEC(create_sa): sa created, (sa) sa_dest= 20.20.20.21, sa_prot= 50, sa_spi= 0xAF62683(183903875), sa_trans= esp-rfc1829 , sa_conn_id= 15

```
Apr 2 12:04:04.619: IPSEC(create_sa): sa created, (sa)
sa_dest= 20.20.20.20, sa_prot= 50, sa_spi=
0x12180818(303564824), sa_trans= esp-rfc1829 , sa_conn_id= 16
!--- The IPsec SAs are created, and ICMP traffic can flow.
```

Commandes show du routeur homologue

```
!--- This illustrates a series of show command output after
!--- IKE/IPsec negotiation takes place. goss-c2-2513#show
crypto isakmp sa dst src state conn-id slot 20.20.20.21
20.20.20.20 QM_IDLE 14 0 goss-c2-2513#show crypto ipsec sa
interface: Serial0 Crypto map tag: armadillo, local addr.
20.20.20.21 local ident (addr/mask/prot/port):
(50.50.50.0/255.255.255.0/0/0) remote ident
(addr/mask/prot/port): (60.60.60.0/255.255.255.0/0/0)
current_peer: 20.20.20.20 PERMIT, flags={origin_is_acl,}
#pkts encaps: 5, #pkts encrypt: 5, #pkts digest 0 #pkts
decaps: 5, #pkts decrypt: 5, #pkts verify 0 #send errors 0,
#recv errors 0 local crypto endpt.: 20.20.20.21, remote
crypto endpt.: 20.20.20.20 path mtu 1500, media mtu 1500
current outbound spi: 12180818 inbound esp sas: spi:
0xAF62683(183903875) transform: esp-rfc1829 , in use settings
={Var len IV, Tunnel, } slot: 0, conn id: 15, crypto map:
armadillo sa timing: remaining key lifetime (k/sec):
(4607999/118) IV size: 8 bytes replay detection support: N
inbound ah sas: outbound esp sas: spi: 0x12180818(303564824)
transform: esp-rfc1829 , in use settings ={Var len IV,
Tunnel, } slot: 0, conn id: 16, crypto map: armadillo sa
timing: remaining key lifetime (k/sec): (4607999/109) IV
size: 8 bytes replay detection support: N outbound ah sas:
goss-c2-2513#show crypto isakmp policy Protection suite of
priority 1 encryption algorithm: DES - Data Encryption
Standard (56 bit keys). hash algorithm: Message Digest 5
authentication method: Pre-Shared Key Diffie-Hellman group:
#1 (768 bit) lifetime: 86400 seconds, no volume limit Default
protection suite encryption algorithm: DES - Data Encryption
Standard (56 bit keys). hash algorithm: Secure Hash Standard
authentication method: Rivest-Shamir-Adleman Signature
Diffie-Hellman group: #1 (768 bit) lifetime: 86400 seconds,
no volume limit goss-c2-2513#show crypto map Crypto Map
"armadillo" 1 ipsec-isakmp Peer = 20.20.20.20 Extended IP
access list 101 access-list 101 permit ip 50.50.50.0
0.0.0.255 60.60.60.0 0.0.0.255 Current peer: 20.20.20.20
Security association lifetime: 4608000 kilobytes/190 seconds
PFS (Y/N): N Transform sets={ MamaBear, PapaBear, BabyBear, }
```

Conseils d'implémentation pour IPsec

Voici quelques conseils d'implémentation pour IPsec :

- Assurez-vous que la connectivité existe entre les points de terminaison de la communication avant que vous ne configuriez le chiffrement.
- Assurez-vous que DNS fonctionne sur le routeur, ou que vous avez entré le nom d'hôte de CA, si vous utilisez une CA.
- IPsec utilise les protocoles IP 50 et 51, et le trafic IKE passe sur le protocole 17, port 500 (UDP 500). Assurez-vous que ceux-ci sont admis de manière convenable.
- Faites attention à ne pas utiliser le mot any dans votre ACL. Ceci pose des problèmes. Référez-vous aux directives d'utilisation pour **access-list** dans la référence de la commande PIX pour plus d'informations.

- Les combinaisons de transformations recommandées sont les suivantes :`esp-des and esp-sha-hmac`
  `ah-sha-hmac and esp-des`
- Rappelez-vous qu'AH est juste un en-tête authentifié. Le flux de données utilisateur en cours n'est pas chiffré. Vous avez besoin d'ESP pour le chiffrement du flux de données. Si vous utilisez seulement AH et envoyez le libellé à travers le réseau, ne soyez pas étonné. Utilisez également ESP si vous utilisez AH. Notez qu'ESP peut également exécuter l'authentification. Par conséquent, vous pouvez utiliser une combinaison de transformation telle que **esp-des** et **esp-sha-hmac**.
- **ah-rfc1828** et **esp-rfc1829** sont des transformations obsolètes incluses pour la rétrocompatibilité des implémentations plus anciennes d'IPsec. Si l'homologue ne prend pas en charge les nouvelles transformations, essayez celles-ci à la place.
- SHA est plus lent et plus sécurisé que MD5, tandis que MD5 est plus rapide et moins sécurisé que ce SHA. Dans quelques communautés, le niveau de confort avec MD5 est très bas.
- En cas de doute, utilisez le tunnel mode. Le tunnel mode est le mode par défaut et il peut être utilisé dans le transport mode, aussi bien pour ses capacités VPN.
- Pour les utilisateurs de chiffrement classique qui mettent à jour le Logiciel Cisco IOS Version 11.3, les méthodes de stockage des commandes de chiffrement dans la configuration ont changé afin de tenir compte d'IPsec. En conséquence, si les utilisateurs de chiffrement classique reviennent au Logiciel Cisco IOS Version 11.2, ces utilisateurs doivent ressaisir leurs configurations de chiffrement.
- Si vous faites un test ping sur une liaison chiffrée quand vous terminez votre configuration, le processus de négociation peut durer un peu de temps, six secondes environ sur un Cisco 4500 et 20 secondes environ sur Cisco 2500, parce que les SA n'ont pas été encore négociées. Quoique tout soit correctement configuré, votre ping peut échouer au commencement. Les commandes **debug crypto ipsec** et **debug crypto isakmp** montrent ce qui se produit. Une fois que vos flux de données chiffrés ont terminé leur configuration, le ping fonctionne très bien.
- Si vous rencontrez des problèmes avec votre négociation et que vous apportez des modifications à la configuration, utilisez les commandes **clear crypto is** et **clear crypto sa** afin de vider les bases de données avant de réessayer. Cela force la négociation à recommencer, sans être gênée par une négociation traditionnelle. Les commandes **clear crypto is** et **clear cry sa** sont très utiles de cette manière.

Aide et liens pertinents

Informations IPsec

- Page d'assistance IPsec
- Stratégies de chiffrement et procédures ECRA - Envoyez un email à export@cisco.com

Autres exemples de configuration pour IPsec

- Configuration et dépannage du chiffrement réseau-couche Cisco : IPSec et ISAKMP
- Aperçu de la sécurité réseau IPsec
- Documentation relative à la configuration IPsec du pare-feu PIXPIX 5.1PIX 5.2PIX 5.3PIX 6.0PIX 6.1PIX 6.2PIX 6.3

Contactez l'assistance technique Cisco au (800) 553-24HR, (408) 526-7209, ou envoyez un email à tac@cisco.com si vous avez besoin davantage d'aide sur IPsec.

Références

Harkins, *spécification fonctionnelle d'unité de logiciel de caractéristique de* D. *ISAKMP/Oakley Protocol.* ENG-0000 Rév A. Cisco Systems.

Madson, *spécification fonctionnelle ENG-17610 Rév F.* Cisco Systems d'*unité de* C. *IPSec Software*.

Kaufman, C. Perlman R. et Spencer, *sécurité des réseaux* M. *: Private Communication in a Public World.* Prentice Hall, 1995.

Schneier, B. *Applied Cryptography : Protocols, Algorithms, and Source Code in C.* Second Ed. John Wiley & Sons, Inc.

Diverses ébauches sur la sécurité IP IETF

Informations connexes

- Page d'assistance IPsec
- Fonctionnement des réseaux VPN
- Solutions de dépannage les plus fréquentes concernant un VPN IPsec LAN à LAN et d'accès à distance
- Support et documentation techniques - Cisco Systems