SNRS

Securing Networks with Cisco Routers and Switches

Volume 2

Version 2.0

Student Guide

Editorial, Production, and Web Services: 02.06.07

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

DISCLAIMER WARRANTY: THIS CONTENT IS BEING PROVIDED “AS IS.” CISCO MAKES AND YOU RECEIVE NO WARRANTIES IN CONNECTION WITH THE CONTENT PROVIDED HEREUNDER, EXPRESS, IMPLIED, STATUTORY OR IN ANY OTHER PROVISION OF THIS CONTENT OR COMMUNICATION BETWEEN CISCO AND YOU. CISCO SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. This learning product may contain early release content, and while Cisco believes it to be accurate, it falls subject to the disclaimer above.

Table of Contents

Volume 2

Module 4

Secured Connectivity

Overview

Remote-access virtual private networks (VPNs) allow secure access to corporate resources by establishing an encrypted tunnel across the Internet. The ubiquity of the Internet, combined with the VPN technologies of today, allows organizations to cost-effectively and securely extend the reach of their networks to anyone, anyplace, anytime. In this module, you will be introduced to IP Security (IPsec) services and generic routing encapsulation (GRE). You will then set up site-to-site and Dynamic Multipoint VPNs (DMVPNs). You will also learn how to set up a portal to accommodate Secure Sockets Layer (SSL) VPN (Cisco IOS WebVPN).

Module Objectives

Upon completing this module, you will be able to implement secure IPsec VPNs and GRE tunnels using Cisco routers. This ability includes being able to meet these objectives:

- Describe some basic characteristics and protocols used in IPsec configurations
- Describe the various types of VPNs available through Cisco IOS Software including site-to-site, DMVPN, Cisco Easy VPN, and Cisco IOS WebVPN (SSL)
- Configure an IPsec site-to-site VPN using pre-shared keys
- Configure an IPsec site-to-site VPN using digital certificates
- Describe and configure a GRE site-to-site tunnel
- Describe and configure a DMVPN
- Describe and configure a Cisco IOS SSL VPN
- Plan, configure, operate, and troubleshoot IPsec VPNs using Cisco Easy VPN

Lesson 1

Introducing IPsec

Overview

In the current business environment, it is critical that corporate networks connected to the Internet offer flexible and secure virtual private network (VPN) access with IP Security (IPsec). Connecting remote sites over the Internet provides a great cost-saving opportunity when compared to the traditional WAN access such as Frame Relay or ATM. With IPsec technology, customers now can build VPNs over the Internet with the security of encryption protection against wire taping or intruding on the private communication. In this lesson, you will be introduced to RFC 2401, *Security Architecture for the Internet Protocol* (defining IPsec), some of the underlying protocols used by IPsec, and the tasks involved with configuration of IPsec on a router.

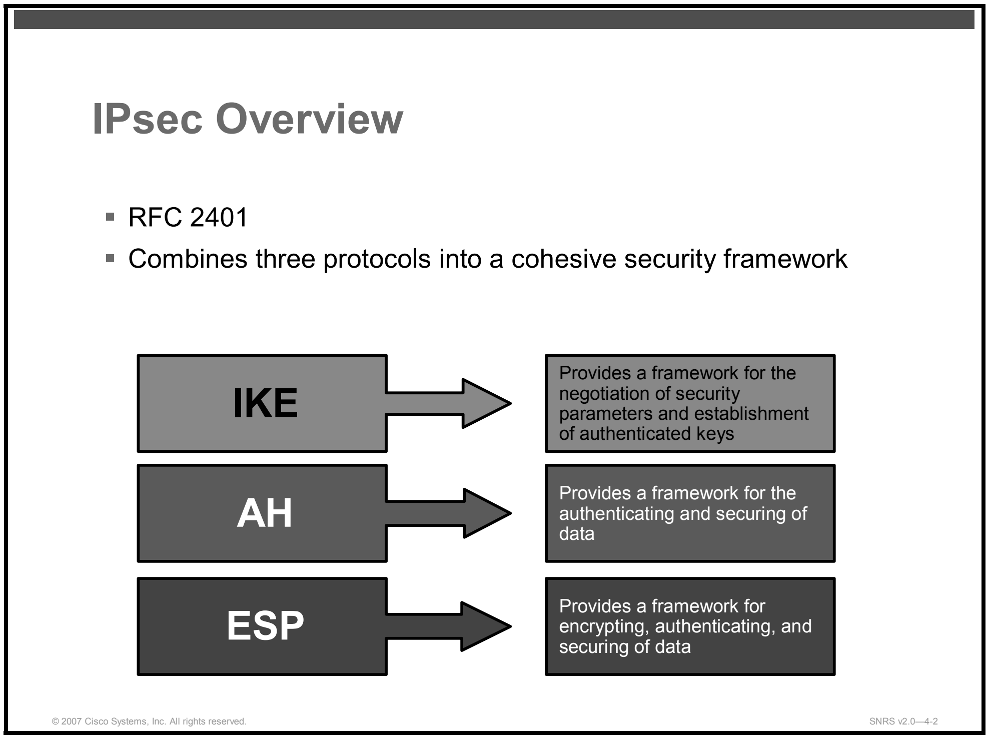
Objectives

Upon completing this lesson, you will be able to describe some basic characteristics and protocols used in IPsec configurations. This ability includes being able to meet these objectives:

- Describe the basic functionality and protocols involved with IPsec VPNs
- Describe AH
- Describe ESP
- Describe the components and characteristics of the IKE protocol
- Describe ISAKMP
- Describes some other protocols and terminologies used with IPsec VPNs
- Describe the tasks required to configure IPsec on a Cisco router

IPsec Overview

This topic describes the basic functionality and protocols involved with IPsec VPNs.



IPsec Overview

- RFC 2401
- Combines three protocols into a cohesive security framework

IKE	Provides a framework for the negotiation of security parameters and establishment of authenticated keys
AH	Provides a framework for the authenticating and securing of data
ESP	Provides a framework for encrypting, authenticating, and securing of data

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-2

IPsec is designed to provide interoperable, high-quality, and cryptographically based security. IPsec is defined in (RFC 2401). The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays, confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and upper-layer protocols (ULPs). Because these services are provided at the IP layer, they can be used by any higher-layer protocol (for example TCP, User Datagram Protocol [UDP], and Border Gateway Protocol [BGP]).

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm (or algorithms) to use for the service (or services), and put in place any cryptographic keys required to provide the requested services. IPsec can be used to protect one or more paths between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

Note The term “security gateway” is predominantly used to refer to an intermediate system that implements IPsec protocols (for example, a router or a firewall implementing IPsec).

The IPsec protocol provides IP network layer encryption and defines a new set of headers to be added to IP datagrams. These new headers are placed after the IP header and before the Layer 4 protocol (typically TCP or UDP). They provide information for securing the payload of the IP packet.

Simply put, IPsec provides secure tunnels between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters that should be used to protect these sensitive packets, by specifying the characteristics of these tunnels. Then, when the IPsec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

These tunnels are sets of security associations (SAs) that are established between two IPsec peers. The SAs define which protocols and algorithms should be applied to sensitive packets and also specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (Authentication Header [AH] or Encapsulating Security Payload [ESP]).

Using IPsec, you will define which traffic should be protected between two IPsec peers by configuring access control lists (ACLs) and then applying these ACLs to interfaces by way of crypto map sets. Therefore, traffic may be selected based on source and destination address and, optionally, Layer 4 protocol and port.

Note The ACLs used for IPsec are used only to determine which traffic should be protected by IPsec, not which traffic should be blocked or permitted through the interface. Separate ACLs define blocking and permitting at the interface.

A crypto map set can contain multiple entries, each with a different ACL. The crypto map entries are searched in order; the router attempts to match the packet to the ACL specified in that entry.

When a packet matches a **permit** entry in a particular ACL and the corresponding crypto map entry is configured as **ipsec-isakmp**, IPsec uses Internet Key Exchange (IKE) to negotiate with the remote peer to set up the necessary IPsec SAs on behalf of the data flow. The negotiation uses information specified in the crypto map entry and the data flow information from the specific ACL entry.

When a packet matches a **permit** entry in a particular ACL and the corresponding crypto map entry is configured as **ipsec-manual**, the SAs are installed via the configuration, without the intervention of IKE. If the SAs do not exist, IPsec did not have all of the necessary pieces configured.

Once established, the set of SAs (outbound, to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the router. “Applicable” packets match the same ACL criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Note If IKE is used to establish the SAs, the SAs will have lifetimes so that they will periodically expire and require renegotiation.

Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams might be just authenticated while other data streams must both be encrypted and authenticated.

ACLs associated with IPsec crypto map entries also represent which traffic the router requires to be protected by IPsec. Inbound traffic is processed against the crypto map entries; if an unprotected packet matches a permit entry in a particular ACL associated with an IPsec crypto map entry, that packet is dropped because it was not sent as an IPsec-protected packet.

Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPsec protected traffic. During the IPsec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

IPsec combines the following security protocols:

- AH
- ESP
- IKE

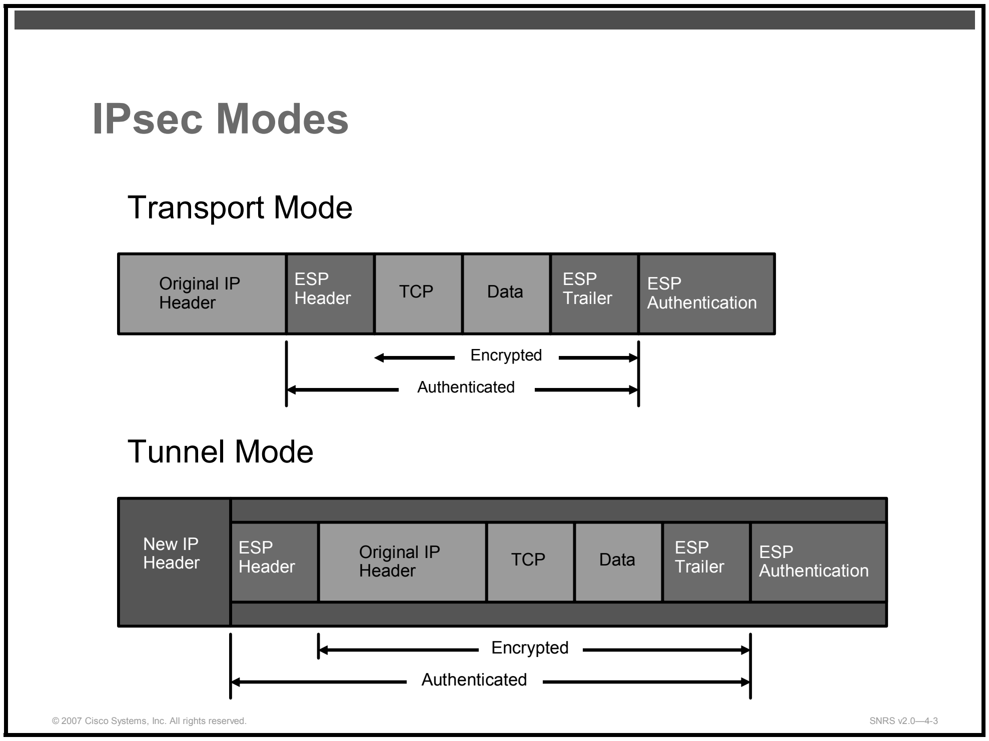
AH and ESP can be used independently or together, although for most applications just one of them is sufficient. For both of these protocols, IPsec does not define the specific security algorithms to use, but rather, provides an open framework for implementing industry-standard algorithms. Initially, most implementations of IPsec support Message Digest 5 (MD5) from RSA Security ("RSA" stands for Rivest, Shamir, and Adleman, the three inventors) or the Secure Hash Algorithm (SHA) as defined by the U.S. government for integrity and authentication. The Data Encryption Standard (DES) is currently the most commonly offered bulk encryption algorithm, although RFCs are available that define how to use many other encryption systems.

Because these security services use shared secret values (cryptographic keys), IPsec relies on a separate set of mechanisms for putting these keys in place.

Security Associations

The concept of an SA is fundamental to IPsec. Both AH and ESP make use of SAs, and a major function of IKE is the establishment and maintenance of SAs. All implementations of AH or ESP *must* support the concept of an SA.

An SA is a simplex connection that affords security services to the traffic carried by it. Security services are afforded to an SA by the use of AH, or ESP, but not both. If both AH and ESP protection is applied to a traffic stream, two (or more) SAs are created to afford protection to the traffic stream. To secure typical, bidirectional communication between two hosts, or between two security gateways, two SAs (one in each direction) are required.



IPsec Modes

Transport Mode

Tunnel Mode

SNRS v2.0—4-3

IPsec Modes

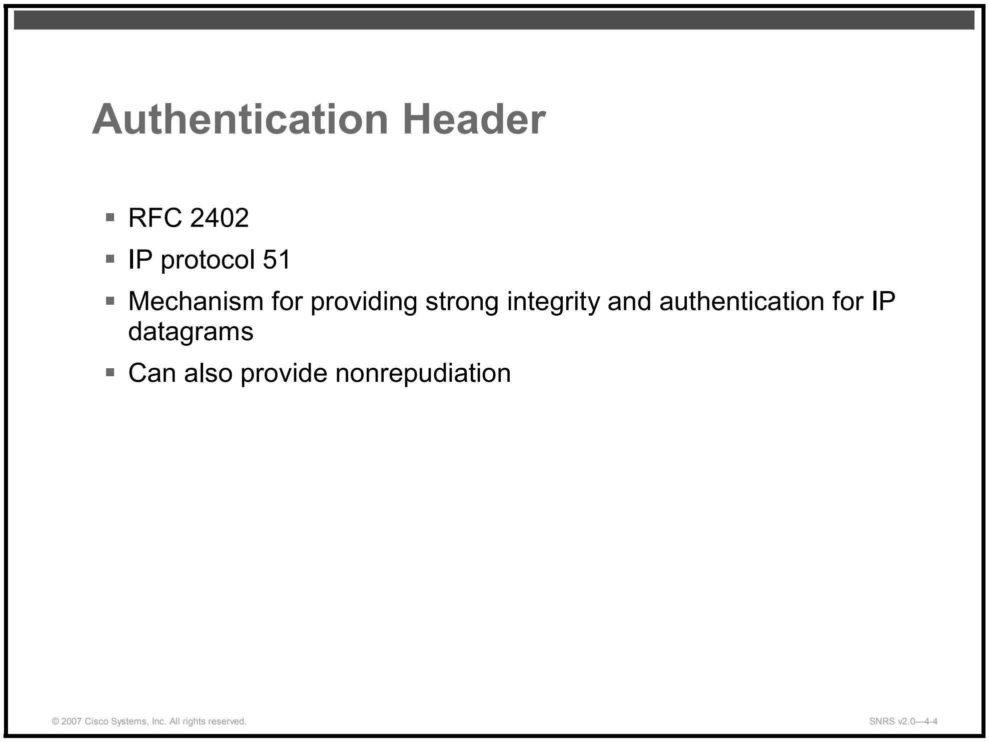
IPsec has two methods of forwarding data across a network: transport mode and tunnel mode. Each differs in their application and in the amount of overhead added to the passenger packet.

Here are some examples:

- **ESP tunnel mode:** Tunnel mode adds an additional new IP header before the ESP header. Tunnel mode encapsulates and protects an entire IP packet. Because tunnel mode encapsulates or hides the original IP header of the packet, a new IP header must be added for the packet to be successfully forwarded. The encrypting routers themselves own the IP addresses used in these new headers. Tunnel mode may be employed with either or both ESP and AH. Using tunnel mode results in additional packet expansion of approximately 20 bytes (B) associated with the new IP header
- **ESP transport mode:** IPsec transport mode inserts an ESP header between the IP header and the transport layer header (TCP header in this example). In this case, transport mode saves an additional IP header, which results in less packet expansion. Transport mode can be deployed with either or both ESP and AH.

Authentication Header

This topic describes the AH protocol.



Authentication Header

- RFC 2402
- IP protocol 51
- Mechanism for providing strong integrity and authentication for IP datagrams
- Can also provide nonrepudiation

SNRS v2.0—4-4

IP AH, a key protocol in the IPsec architecture, is used to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replays. This protection service against replay is an optional service to be selected by the receiver when an SA is established.

AH is defined in RFC 2402, *IP Authentication Header.* The IP version 4 (IPv4) or IP version 5 (IPv6) header immediately preceding the AH will contain the value 51 in its Next Header (or Protocol) field.

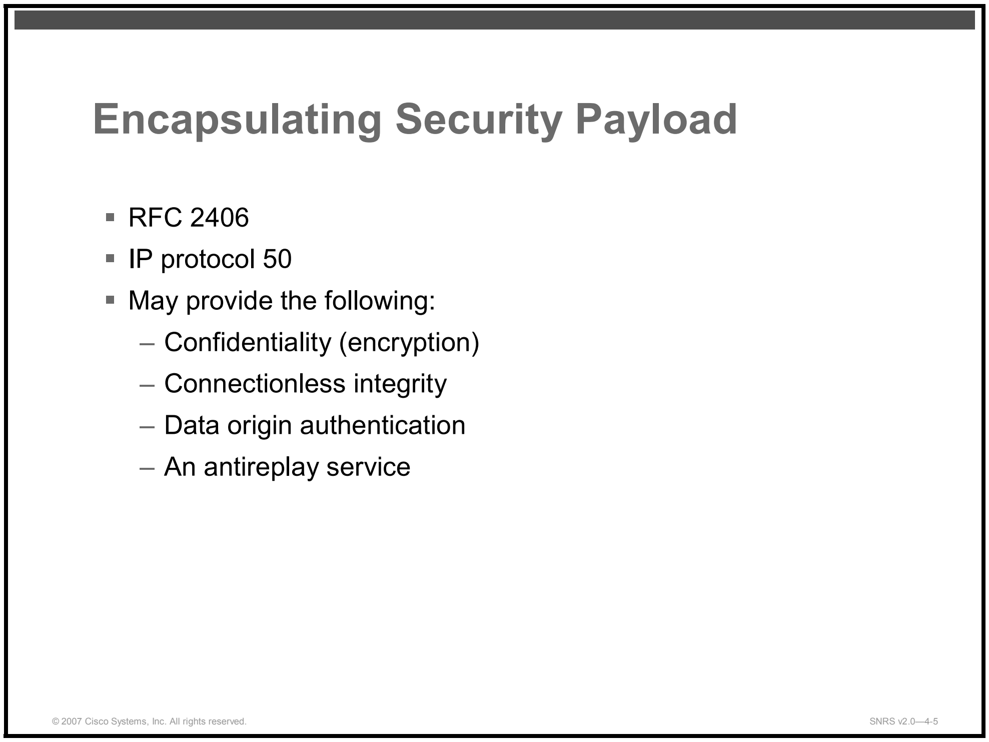
The AH is a mechanism for providing strong integrity and authentication for IP datagrams. Confidentiality and protection from traffic analysis are not provided by the AH. Users who need confidentiality should consider using ESP, either in lieu of or in conjunction with the AH.

The primary difference between the authentication provided by ESP and AH is the extent of the coverage. Specifically, ESP does not protect any IP header fields unless those fields are encapsulated by ESP.

Like ESP, AH may be employed in two ways: transport mode or tunnel mode.

Encapsulating Security Payload

This topic describes the ESP protocol.



Encapsulating Security Payload

- RFC 2406
- IP protocol 50
- May provide the following:
  - Confidentiality (encryption)
  - Connectionless integrity
  - Data origin authentication
  - An antireplay service

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-5

ESP is designed to provide a mix of security services in IPv4 and IPv6. ESP seeks to provide confidentiality and integrity by encrypting data to be protected and placing the encrypted data in the data portion of the IP ESP payload.

ESP is defined in RFC 2406, *IP Encapsulating Security Payload (ESP).*

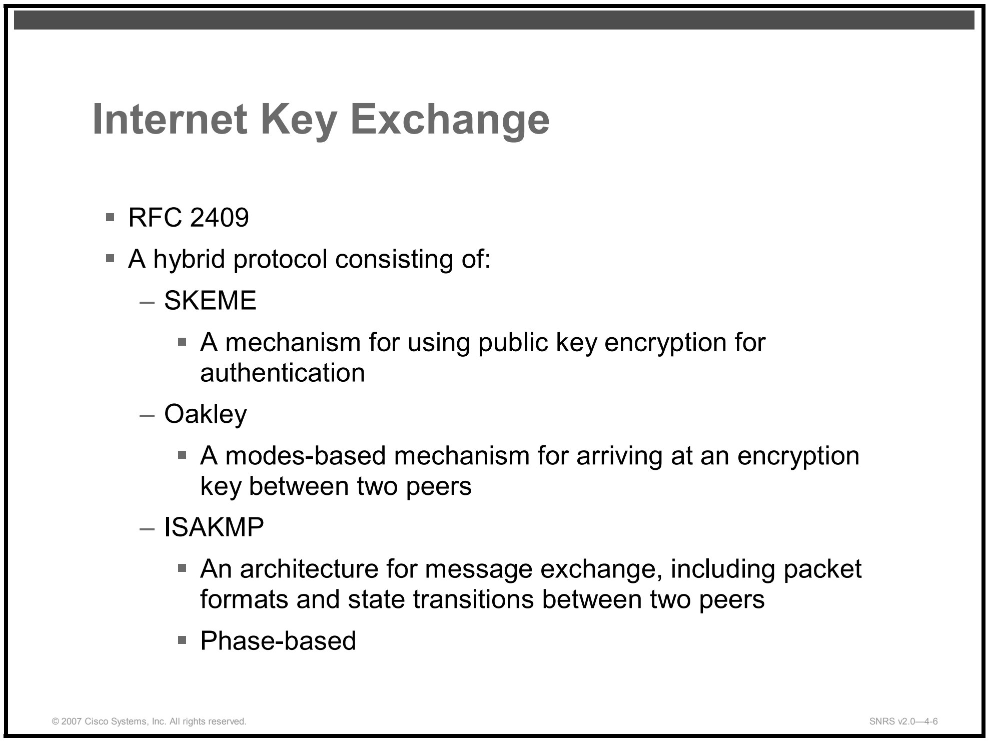
— IP protocol 50

Note Use of ESP will increase the IP protocol processing costs in participating systems and will also increase the communications latency. The increased latency is primarily due to the encryption and decryption required for each IP datagram containing an ESP.

The ESP header is inserted after the IP header and before the ULP header (transport mode) or before an encapsulated IP header (tunnel mode). The Internet Assigned Numbers Authority (IANA) has assigned IP protocol 50 to ESP. The header immediately preceding an ESP header always contains the value 50 in its Next Header (IPv6) or Protocol (IPv4) field. ESP consists of an unencrypted ESP header followed by encrypted data. The encrypted data includes both the protected ESP header fields and the protected user data, which is either an entire IP datagram or a ULP frame (such as TCP or UDP).

Internet Key Exchange

This topic describes the IKE protocol.



Internet Key Exchange

- RFC 2409
- A hybrid protocol consisting of:
  - SKEME
    - A mechanism for using public key encryption for authentication
  - Oakley
    - A modes-based mechanism for arriving at an encryption key between two peers
  - ISAKMP
    - An architecture for message exchange, including packet formats and state transitions between two peers
    - Phase-based

SNRS v2.0—4-6

IKE is a hybrid protocol that uses part Oakley and part of another protocol suite called Skeme inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. IKE is used to establish a shared security policy and authenticated keys for services (such as IPsec) that require keys. Before any IPsec traffic can be passed, each router, firewall, and host must be able to verify the identity of its peer. This can be done by manually entering pre-shared keys into both hosts, by a certificate authority (CA) service, or the forthcoming Secure Domain Name System (DNSSEC). IKE is the protocol formerly known as ISAKMP/Oakley, and is defined in RFC 2409, *The Internet Key Exchange (IKE).* A potential point of confusion is that the acronyms “ISAKMP” and “IKE” are both used in Cisco IOS Software to refer to the same thing. These two items are somewhat different.

IKE uses a Diffie-Hellman (DH) key exchange to set up a shared session secret, from which cryptographic keys are derived.

IKE is a hybrid solution that uses the following:

- **Skeme:** Describes a versatile key exchange technique that provides anonymity, reputability, and quick key refreshment
- **Oakley:** Describes a series of key exchanges called “modes” and details the services provided by each (for example, perfect forward secrecy for keys, identity protection, and authentication)
- **ISAKMP:** Provides a framework for authentication and key exchange but does not define them; designed to support many different key exchanges

Why IKE?

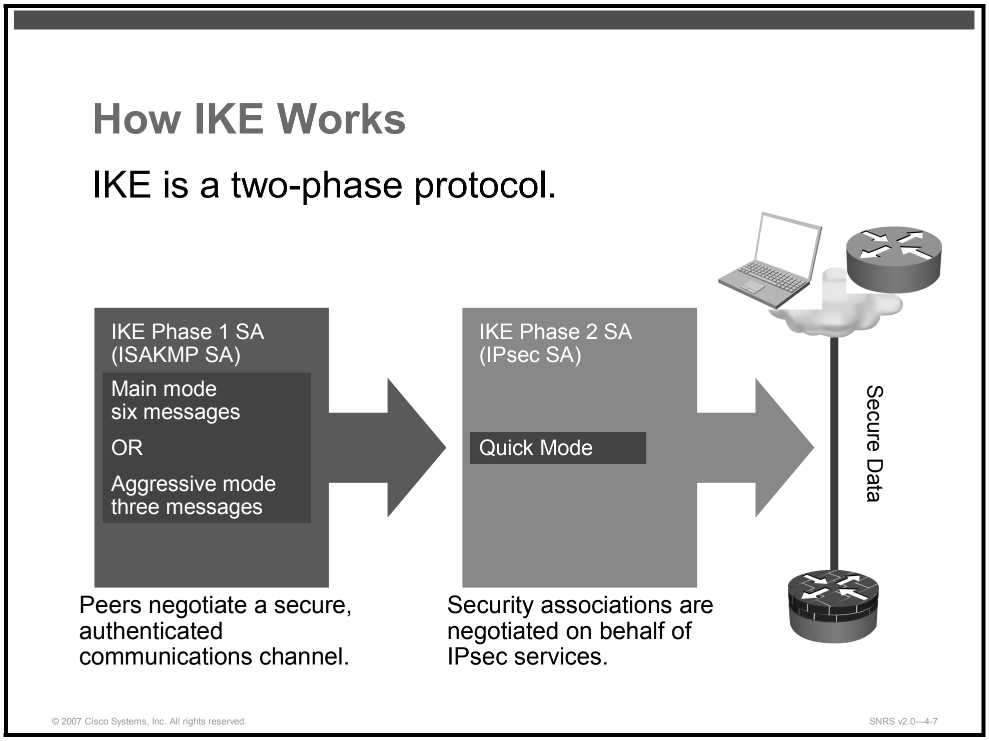
IKE solves the enormous problem of a manual and unscalable implementation of IPsec by automating the entire key exchange process.

Here are some of the reasons for implementing IKE in your IPsec configuration:

- Scalability
- Manageable manual configuration
- Negotiates SA characteristics
- Automatic key generation
- Automatic key refresh

How IKE Works

This section covers the operation of IKE.



How IKE Works

IKE is a two-phase protocol.

IKE Phase 1 SA (ISAKMP SA)

Main mode six messages

OR

Aggressive mode three messages

IKE Phase 2 SA (IPsec SA)

Quick Mode

Secure Data

Peers negotiate a secure, authenticated communications channel.

Security associations are negotiated on behalf of IPsec services.

SNRS v2.0—4-7

Oakley and Skeme each define a method to establish an authenticated key exchange. This includes the construction of payloads, the information that payloads carry, the order in which payloads are processed, and how they are used.

While Oakley defines modes, ISAKMP defines phases. The relationship between the two is very straightforward, and IKE presents different exchanges as modes that operate in one of two phases.

IKE Phase 1

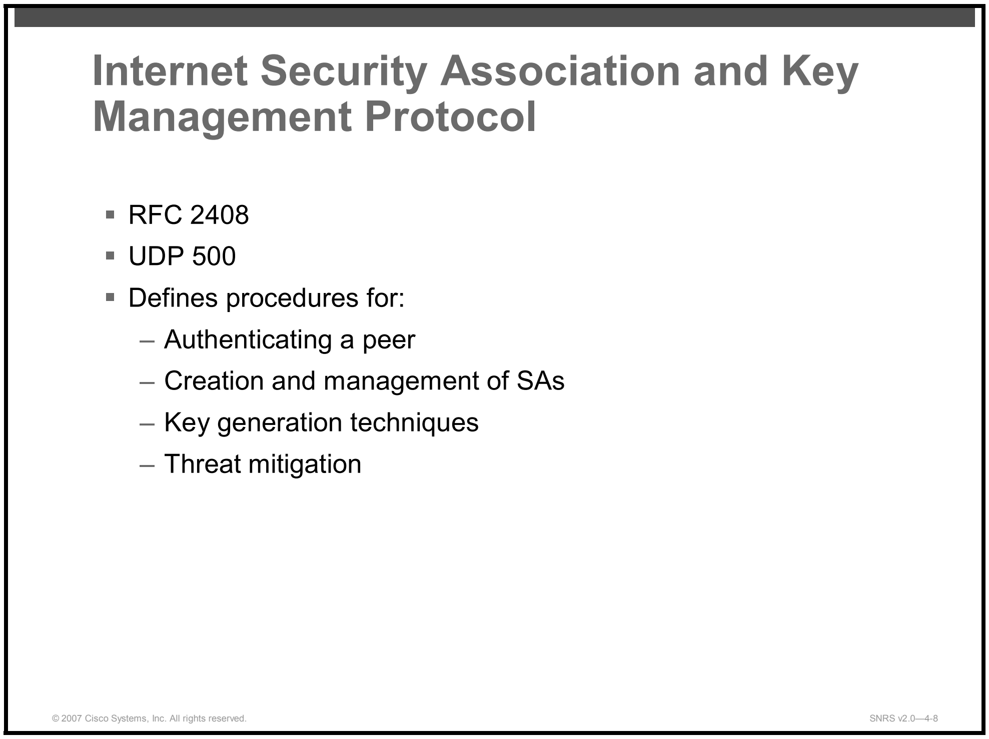
The two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP SA. Main mode and aggressive mode each accomplish an IKE Phase 1 exchange. Main mode and aggressive mode *must only* be used in IKE Phase 1.

IKE Phase 2

SAs are negotiated on behalf of services such as IPsec or any other service that needs key material or parameter negotiation, or both. Quick mode accomplishes an IKE Phase 2 exchange. Quick mode *must only* be used in IKE Phase 2.

Internet Security Association and Key Management Protocol

This topic describes ISAKMP.



Internet Security Association and Key Management Protocol

- RFC 2408
- UDP 500
- Defines procedures for:
  - Authenticating a peer
  - Creation and management of SAs
  - Key generation techniques
  - Threat mitigation

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-8

ISAKMP is defined in RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*.

ISAKMP defines the procedures for authenticating a communicating peer, creation and management of SAs, key generation techniques, and threat mitigation (for example, denial of service [DoS] and replay attacks).

While IPsec is the actual protocol that protects the IP datagrams, ISAKMP is the protocol that negotiates policy and provides a common framework for generating keys that IPsec peers share. ISAKMP does not specify any details of key management or key exchange and is not bound to any key generation technique. Inside of ISAKMP, Cisco uses Oakley for the key exchange protocol. Oakley allows you to choose between different well-known DH groups. Cisco IOS Software supports group 1 (a 768-bit key), group 2 (a 1024-bit key), and group 5 (a 1536-bit key).

ISAKMP and Oakley create an authenticated, secure tunnel between two entities, and then negotiate the SA for IPsec. This process requires that the two entities authenticate themselves to each other and establish shared keys.

Both parties must be authenticated to each other. ISAKMP and Oakley support multiple authentication methods. The two entities must agree on a common authentication protocol through a negotiation process using either RSA signatures, RSA encrypted nonces, or pre-shared keys.

Both RSA signatures and RSA-encrypted nonces authentication require the public key of the remote peer and they also require the remote peer to have your local public key. Public keys are exchanged in ISAKMP in the form of digital certificates. These certificates can be obtained by enrolling in the CA.

Both parties must have a shared session key to encrypt the ISAKMP-Oakley tunnel. The DH protocol is used to agree on a common session key. The exchange is authenticated as described previously to guard against man-in-the-middle attacks.

These two steps, authentication and key exchanges, create the ISAKMP-Oakley SA, which is a secure tunnel between the two devices. One side of the tunnel offers a set of algorithms; the other side must then accept one of the offers or reject the entire connection. When the two sides have agreed on which algorithms to use, they must derive key material to use for IPsec with AH, ESP, or both.

IPsec uses a different shared key than ISAKMP and Oakley. The IPsec shared key can be derived by using DH again to ensure perfect forward secrecy, or by refreshing the shared secret derived from the original DH exchange that generated the ISAKMP-Oakley SA by hashing it with pseudorandom numbers (nonces). The first method provides greater security but is slower. In most implementations, a combination of the two methods is used. That is, DH is used for the first key exchange, and then local policy dictates when to use DH or merely a key refresh. After this is complete, the IPsec SA is established.

Both RSA signatures and RSA-encrypted nonces require the public key of the remote peer, and they also require the remote peer to have your local public key. Public keys are exchanged in ISAKMP in the form of certificates. These certificates are obtained by enrolling in the CA. Currently, if there is no certificate in the router, ISAKMP does not negotiate the protection suite RSA signatures.

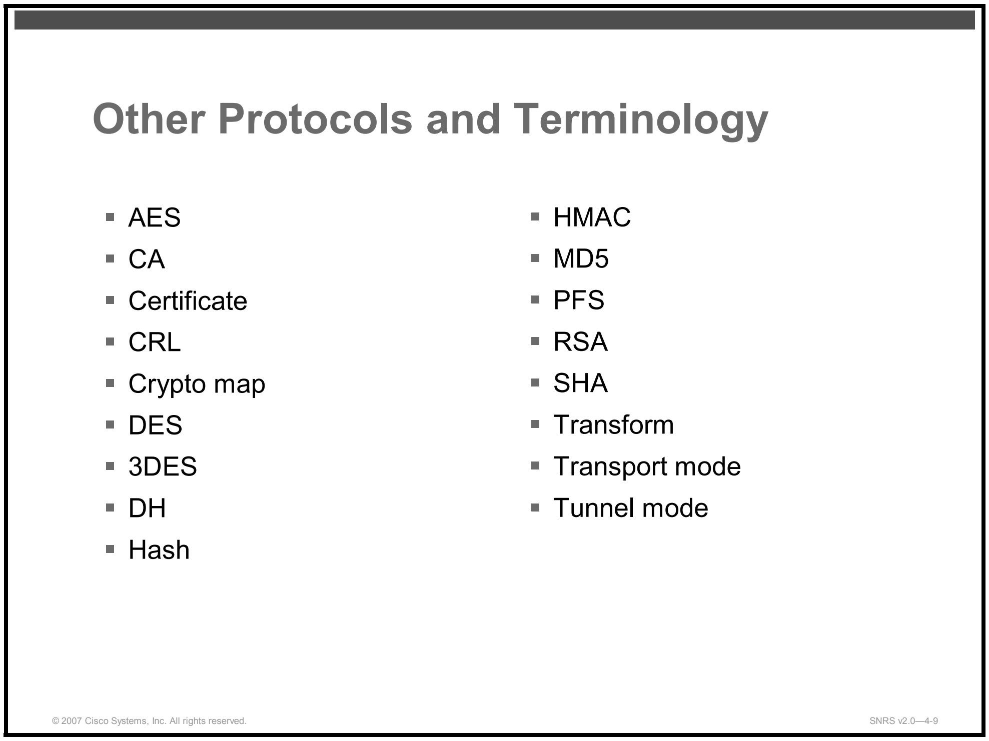
Cisco routers do not create certificates. Routers create keys and request certificates for those keys. The certificates, which bind the keys of the routers to their identities, are created and signed by CAs. This is an administrative function, and the CA always requires some sort of verification that the users are who they say they are. This means that you cannot just create new certificates on the fly.

The communicating machines exchange pre-existing certificates that they have obtained from CAs. The certificates themselves are public information, but the corresponding private keys must be available to anybody who wants to use a certificate to prove identity. However, the private keys also must be kept secret from anybody who should not be able to use that identity.

A certificate may identify a user or a machine. It depends on the implementation. Most early systems probably use a certificate to identify a machine. If a certificate identifies a user, the private key corresponding to that certificate has to be stored in such a way that another user on the same machine cannot use it. That generally means that either the key is kept encrypted, or that the key is kept in a smart card. The encrypted key case is likely to be more common in early implementations. In either case, the user generally has to enter a pass phrase whenever a key is activated.

Other Protocols and Terminology

This topic describes some other protocols and terminologies used with IPsec.



Other Protocols and Terminology

- AES
- CA
- Certificate
- CRL
- Crypto map
- DES
- 3DES
- DH
- Hash
- HMAC
- MD5
- PFS
- RSA
- SHA
- Transform
- Transport mode
- Tunnel mode

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-9

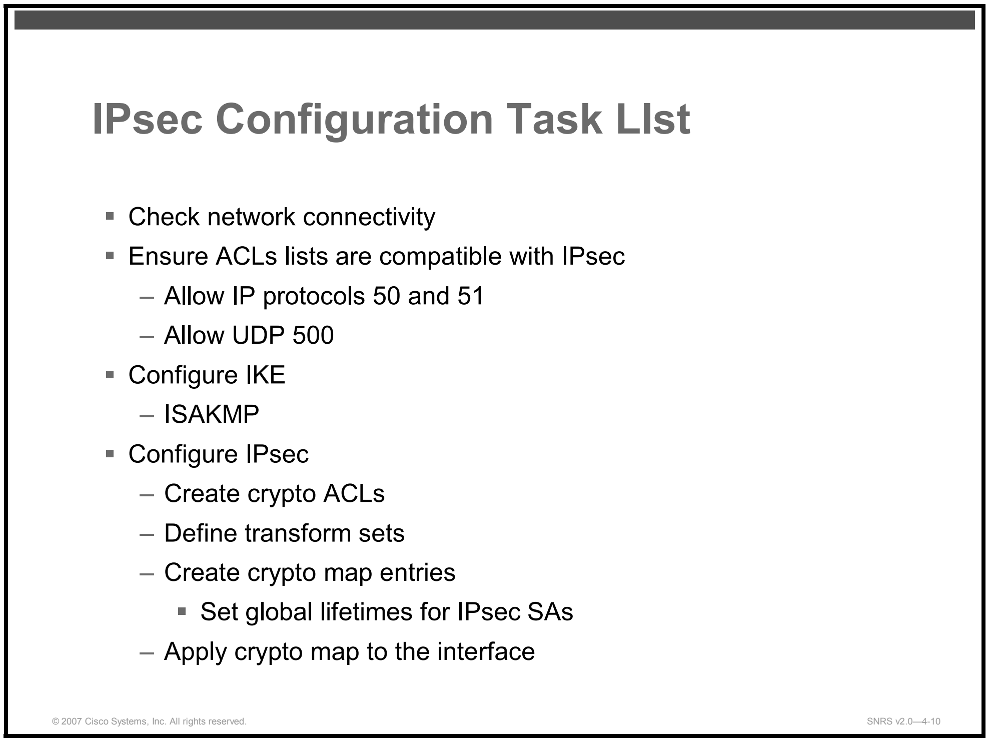
Listed here are some other protocols and terms used with IPsec.

- **Advanced Encryption Standard (AES):** AES was finalized as a Federal Information Processing Standard (FIPS)-approved cryptographic algorithm to be used to protect electronic data transmission (FIPS PUB 197). AES is based on the Rijndael algorithm, which specifies how to use keys with a length of 128, 192, or 256 bits to encrypt blocks with a length of 128, 192, or 256 bits (all 9 combinations of key length and block length are possible).
- **CA:** A CA is a third-party entity with the responsibility to issue and revoke certificates. Each device that has its own certificate and public key of the CA can authenticate every other device within the domain of a given CA. This term also applies to server software that provides these services.
- **Certificate:** A certificate is a cryptographically signed object that contains an identity and a public key associated with this identity.
- **Certificate revocation list (CRL):** A CRL is a digitally signed message that lists all of the current but revoked certificates listed by a given CA.
- **Crypto map:** A crypto map is a Cisco IOS Software configuration entity that performs two primary functions. First, it selects data flows that need security processing. Second, it defines the policy for these flows and the crypto peer that traffic needs to go to. A crypto map is applied to an interface.
- **DES:** The DES was published in 1977 by the National Bureau of Standards (NBS) (the former name of the National Institute of Standards and Technology [NIST]) and is a secret key encryption scheme based on the Lucifer algorithm from IBM. The contrast of DES is public key. Cisco uses DES in classic cryptography, IPsec cryptography, and on the Cisco ASA.

- **Triple DES (3DES):** This is a mode of the DES encryption algorithm that encrypts data three times. Three 64-bit keys are used, instead of one 64-bit key, for an overall key length of 192 bits; the first encryption is encrypted with a second key, and the resulting cipher text is again encrypted with a third key).
- **DH:** This is a method of establishing a shared key over an insecure medium. DH is a component of Oakley.
- **Hash:** This is a one-way function that takes an input message of arbitrary length and produces a fixed-length digest. Cisco uses both SHA and MD5 hashes within its implementation of the IPsec framework.
- **Hashed Message Authentication Code (HMAC): HMAC is** a mechanism for message authentication using cryptographic hashes such as SHA and MD5.
- **MD5:** MD5 is a one-way hashing algorithm that produces a 128-bit hash. Both MD5 and SHA are variations on Message Digest 4 (MD4), which is designed to strengthen the security of this hashing algorithm. SHA is more secure than MD4 and MD5.
- **Perfect forward secrecy (PFS):** PFS ensures that a given IPsec SA key was not derived from any other secret (like some other keys). In other words, if someone breaks a key, PFS ensures that the attacker is not able to derive any other key. If PFS is not enabled, someone can potentially break the IKE SA secret key, copy all the IPsec protected data, and then use knowledge of the IKE SA secret to compromise the IPsec SA setup by this IKE SA. With PFS, breaking IKE does not give an attacker immediate access to IPsec. The attacker needs to break each IPsec SA individually. The Cisco IOS IPsec implementation uses PFS group 1 (DH 768 bit) by default.
- **RSA:** RSA is a public key cryptographic algorithm (named after its inventors, Rivest, Shamir and Adleman) with a variable key length. The main weakness of RSA is that it is significantly slow to compute compared to popular secret key algorithms, such as DES. The Cisco IKE implementation uses a DH exchange to get the secret keys. This exchange can be authenticated with RSA (or pre-shared keys). With the DH exchange, the DES key never crosses the network (not even in encrypted form), which is not the case with the RSA encrypt and sign technique. RSA is not a public domain and must be licensed from RSA Security.
- **SHA:** This is a one-way hash put forth by the NIST. SHA is closely modeled after MD4 and produces a 160-bit digest. Because SHA produces a 160-bit digest, it is more resistant to brute-force attacks than 128-bit hashes (such as MD5), but it is slower.
- **Transform:** A transform describes a security protocol (AH or ESP) with its corresponding algorithms (for example, ESP with the DES cipher algorithm and HMAC and SHA for authentication).
- **Transport mode:** This is an encapsulation mode for AH and ESP. Transport mode encapsulates the upper-layer payload (such as TCP or UDP) of the original IP datagram. This mode can only be used when the peers are the endpoints of the communication. Transport mode is in contrast to tunnel mode.
- **Tunnel mode:** This is an encapsulation of the complete IP datagram for IPsec. Tunnel mode is used to protect datagrams sourced from or destined to non-IPsec systems (such as in a VPN scenario).

IPsec Configuration Task List

This topic describes the tasks required to configure IPsec on a Cisco router.



IPsec Configuration Task LIst

- Check network connectivity
- Ensure ACLs lists are compatible with IPsec
  - Allow IP protocols 50 and 51
  - Allow UDP 500
- Configure IKE
  - ISAKMP
- Configure IPsec
  - Create crypto ACLs
  - Define transform sets
  - Create crypto map entries
    - Set global lifetimes for IPsec SAs
  - Apply crypto map to the interface

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-10

IPsec configuration on a Cisco router involves the configuration of IKE policies and IPsec configurations. You also need to make sure that your network devices are not interfering with the IPsec process.

The tasks discussed here are required to configure IPsec.

Ensure ACLs Are Compatible with IPsec

It is possible to overlook the obvious when adding VPNs to an existing network. Make sure that the router does not have an ACL that is blocking ISAKMP (UDP 500), AH (IP protocol 51), or ESP (IP protocol 50).

Configure ISAKMP

The only reason that IKE exists is to establish SAs for IPsec. IKE must first negotiate an SA (an ISAKMP SA) relationship with the peer before it can establish the IPsec SA. Because IKE negotiates its own policy, it is possible to configure multiple policy statements with different configuration statements, and then let the two hosts come to an agreement.

There are currently two methods used to configure ISAKMP.

- **Pre-shared keys:** Simple, not very scalable

1. Configure ISAKMP protection suite (or suites)

    - Specify what size modulus to use for DH calculation

        - Group 1: 768 bits
        - Group 2: 1024 bits
        - Group 5: 1536 bits.................................
    - Specify a hashing algorithm (MD5 of SHA)
    - Specify the lifetime of the SA (in seconds)
    - Specify the authentication method
        - Pre-shared key
        - RSA encryption
        - RSA signature

2. Configure the ISAKMP key when using pre-shared key for authentication (specify ISAKMP key and peer)

- **Using a CA server:** RSA signatures, RSA-encrypted nonce, scalable throughout an enterprise

1. Create an RSA key for the router
2. Request certificate of the CA
3. Enroll certificates for the client router
4. Configure ISAKMP protection suite (or suites): Specify **rsa-sig** or **rsa-encr** as authentication

Configure IPsec

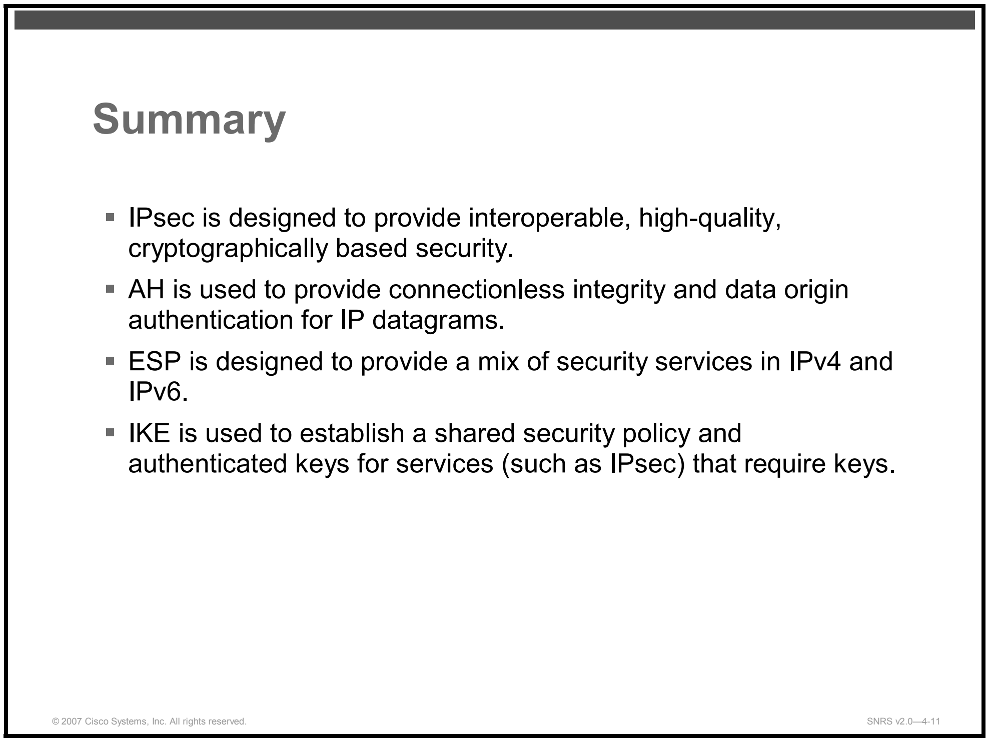
After setting up IKE, you must still setup IPsec. The steps required for IPsec configuration do not rely at all on the IKE configuration method.

When you configure IPsec, you will do the following:

- Create an extended ACL (determines what traffic should be protected by IPsec)
- Create IPsec transform (or transforms).
    - Transform sets are offered to the peer which will choose one.
    - ah-md5-hmac, esp-des, etc....
- Create crypto map (or maps)
    - Specify peer (or peers)
    - Specify SA lifetime
    - Specify transform sets created earlier
    - Specify ACL to match for relevant traffic
- Apply crypto map to an interface (apply the crypto map to the egress interface, not the ingress interface)

Summary

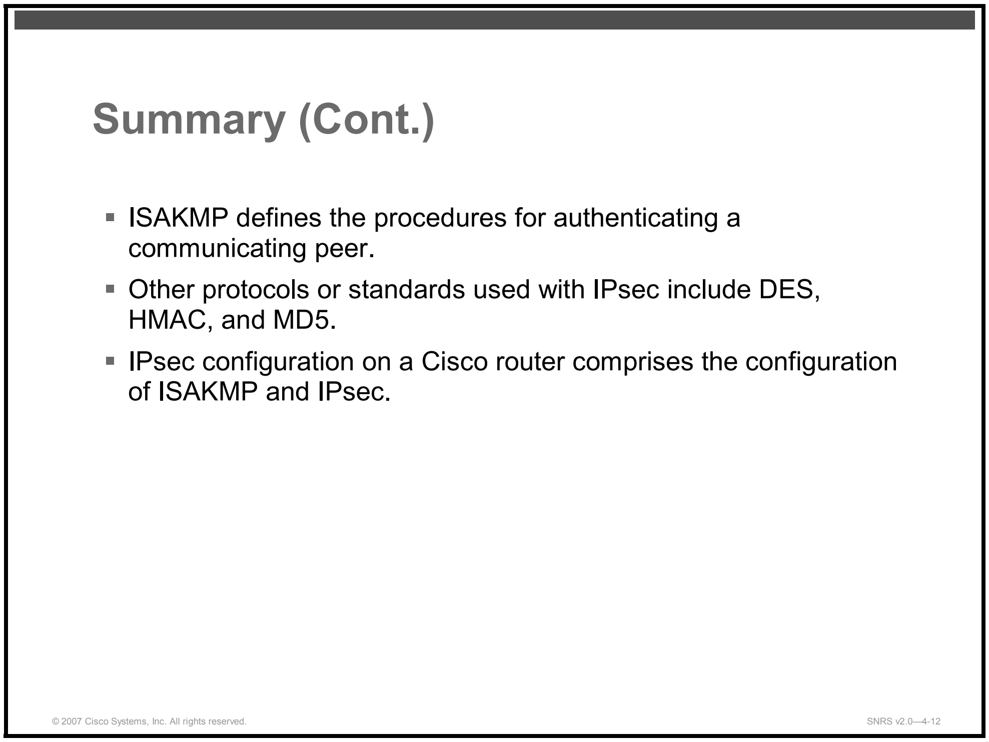
This topic summarizes the key points that were discussed in this lesson.



Summary

- IPsec is designed to provide interoperable, high-quality, cryptographically based security.
- AH is used to provide connectionless integrity and data origin authentication for IP datagrams.
- ESP is designed to provide a mix of security services in IPv4 and IPv6.
- IKE is used to establish a shared security policy and authenticated keys for services (such as IPsec) that require keys.

SNRS v2.0—4-11



Summary (Cont.)

- ISAKMP defines the procedures for authenticating a communicating peer.
- Other protocols or standards used with IPsec include DES, HMAC, and MD5.
- IPsec configuration on a Cisco router comprises the configuration of ISAKMP and IPsec.

SNRS v2.0—4-12

IPsec combines AH, ESP, and IKE to create a framework for secured connectivity.

References

For additional information, refer to these resources:

- RFC 2401, *Security Architecture for the Internet Protocol.*
- RFC 2402, *IP Authentication Header.*
- RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH.*
- RFC 2406, *IP Encapsulating Security Payload (ESP).*
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*.
- RFC 2409, *The Internet Key Exchange (IKE).*
- *Configuring IPsec Network Security*. http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7b1.html.
- Cisco Systems, Inc. Deploying IPsec Virtual Private Network.*:* http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns142/networking_solutions_white_paper09186a0080117919.shtml.
- *Cisco IOS Security Configuration Guide, Release 12.4:* http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_book09186a008043360a.html

Lesson 2

Examining Cisco IOS VPNs

Overview

This lesson will provide you with multiple designs for the implementation of IP Security (IPsec) virtual private network (VPN) configurations over the public Internet infrastructure. The IPsec VPN configurations presented in this lesson will include site-to-site and remote-access VPNs.

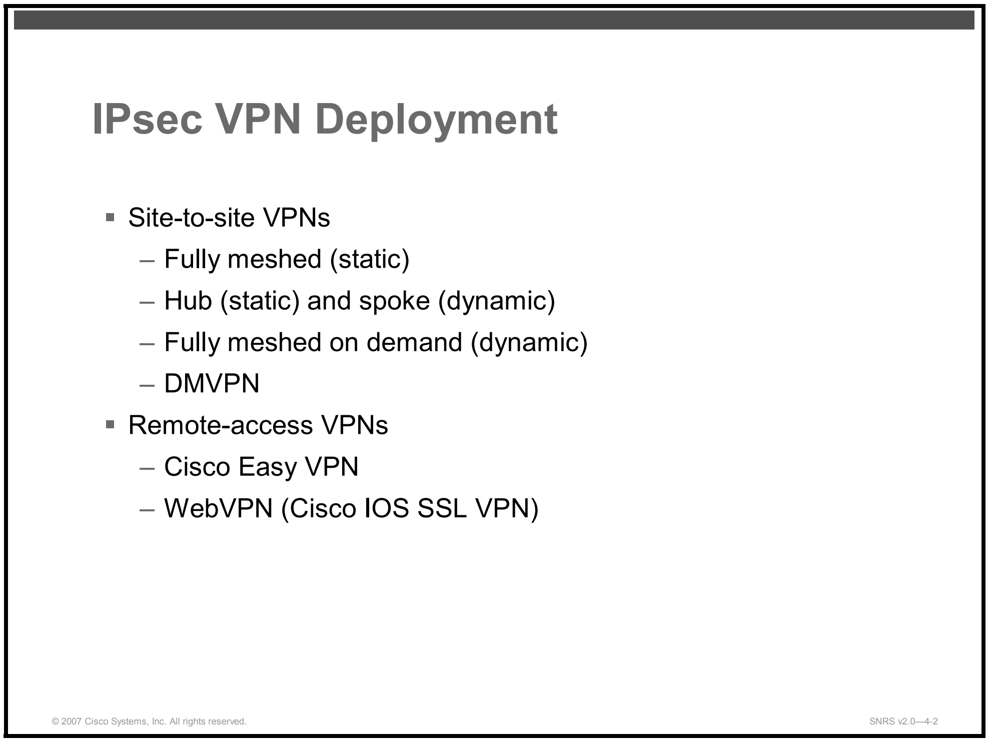
Objectives

Upon completing this lesson, you will be able to describe the various types of VPNs available through Cisco IOS Software including site-to-site and remote-access VPNs. This ability includes being able to meet these objectives:

- Describe various IPsec VPN deployment options
- Describe a fully meshed IPsec VPN
- Describe a hub-and-spoke IPsec VPN
- Describe a DMVPN
- Describe Cisco Easy VPN
- Describe WebVPN

IPsec VPN Deployment Options

This topic describes various IPsec VPN deployment options.



IPsec VPN Deployment

- Site-to-site VPNs
  - Fully meshed (static)
  - Hub (static) and spoke (dynamic)
  - Fully meshed on demand (dynamic)
  - DMVPN
- Remote-access VPNs
  - Cisco Easy VPN
  - WebVPN (Cisco IOS SSL VPN)

SNRS v2.0—4-2

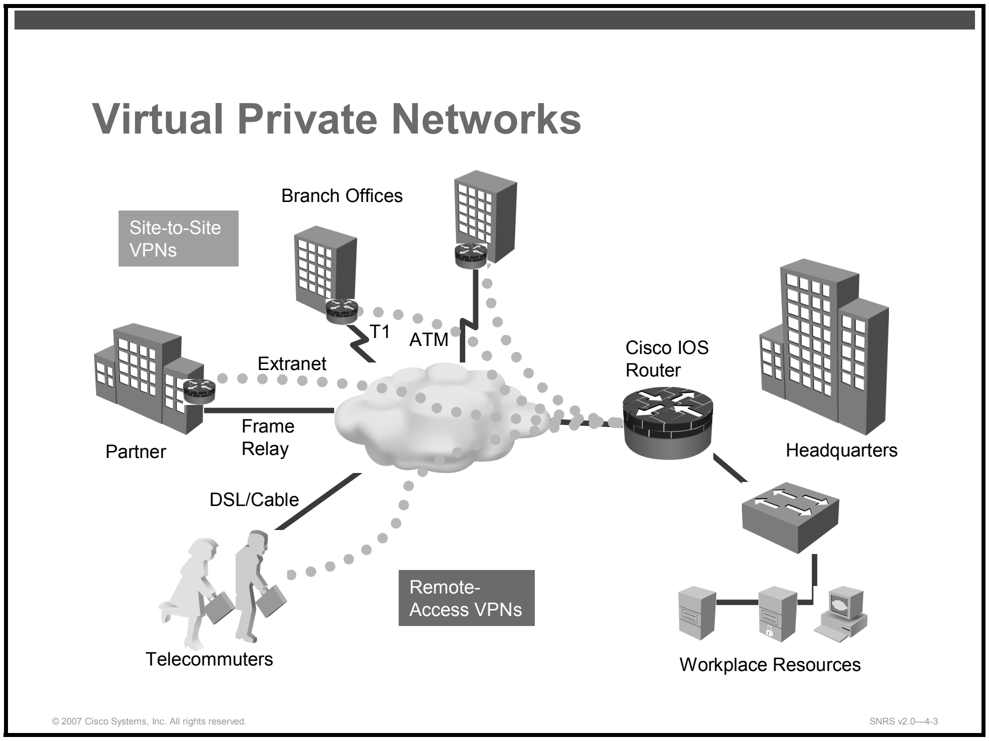
An IPsec VPN is a VPN that is deployed on a shared infrastructure using IPsec encryption technology. IPsec VPNs are used as an alternative to WAN infrastructure that replace or augment existing private networks that utilize leased-line or enterprise-owned Frame Relay and ATM networks. IPsec VPNs do not inherently change WAN requirements, such as support for multiple protocols, high reliability, and extensive scalability, but instead meet these requirements more cost-effectively and with greater flexibility.

Here are some of the deployment options:

- Site to site
  - Fully meshed
  - Hub and spoke
  - Dynamic Multipoint VPN (DMVPN) (hub and spoke or virtual full mesh)
- Remote access
  - Cisco Easy VPN
  - Cisco IOS SSL VPN (WebVPN)

VPN Components

This figure shows some of the VPN components.



Virtual Private Networks

SNRS v2.0—4-3

An IPsec VPN utilizes the most pervasive transport technologies available today (the public Internet, IP backbones, and Frame Relay and ATM networks). The equipment deployed at the edge of the enterprise network and feature integration across the WAN primarily defines the functionality of an IPsec VPN, rather than definitions by the WAN transport protocol.

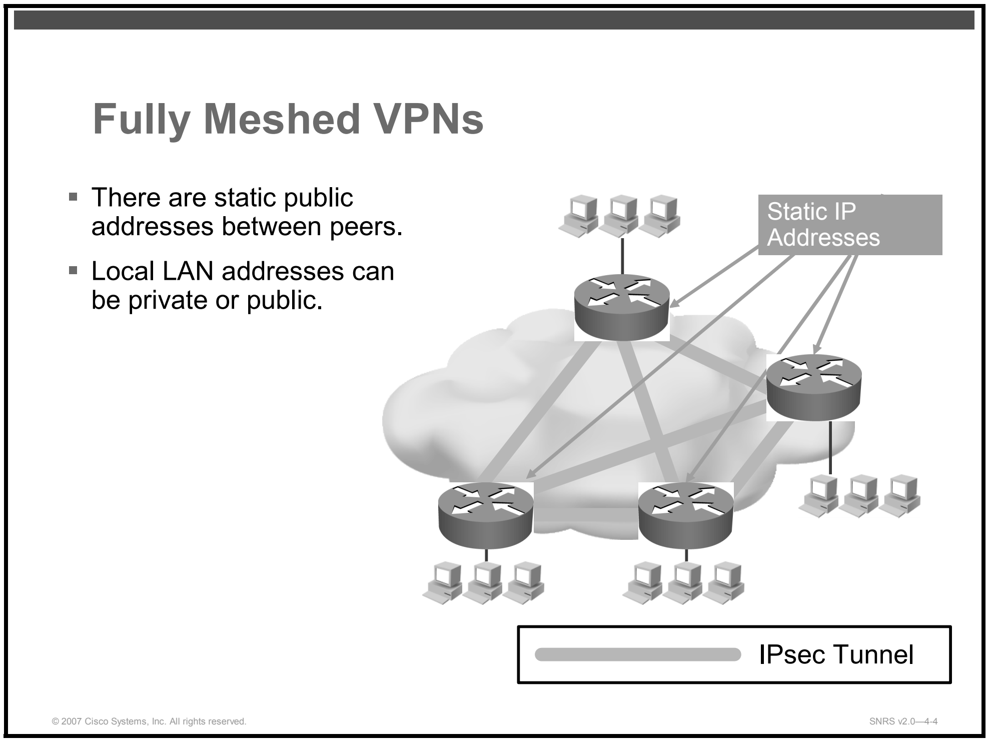
IPsec VPNs are deployed to ensure secure connectivity between the VPN sites. The VPN sites can be either a subnet or a host residing behind routers.

Here are some of the key components of a VPN connection:

- Cisco VPN routers serving as VPN headend termination devices at a central campus (headquarters)
- Cisco VPN access routers serving as VPN branch-end termination devices at the branch office locations
- IPsec and generic routing encapsulation (GRE) tunnels that interconnect the headend and branch-end devices in the VPN
- VPN clients for remote users
- An Internet service provider (ISP) serving as the WAN interconnection medium

Fully Meshed IPsec VPNs

This topic describes fully meshed IPsec VPNs.



Fully Meshed VPNs

- There are static public addresses between peers.
- Local LAN addresses can be private or public.

© 2007 Cisco Systems, Inc. All rights reserved.   SNRS v2.0—4-4

The fully meshed site-to-site design refers to a mesh of IPsec tunnels connecting between remote sites. For any-to-any connectivity, a full mesh of tunnels is required to provide a path between all of the sites. Site-to-site VPNs are primarily deployed to connect branch office locations to the central site of an enterprise and to each other.

This configuration requires the IPsec peers to utilize public IP addresses to establish the IPsec tunnels. The public IP addresses are specified in the IPsec peers configuration and require that the public addresses of the VPN routers be static addresses. The VPN site addresses, however, could be private or public addresses, because the site traffic is encrypted before entering the IPsec tunnels.

Characteristics

Here are some of the common characteristics of a fully meshed VPN:

- There are public IP addresses between peers.
- Local LAN addressing may be either public or private.

Benefits

Here are some of the benefits of a fully meshed VPN topology:

- Increased data and network security
- Reduced WAN costs and increased WAN flexibility
- Simple and flexible design and configuration procedure for adding new sites

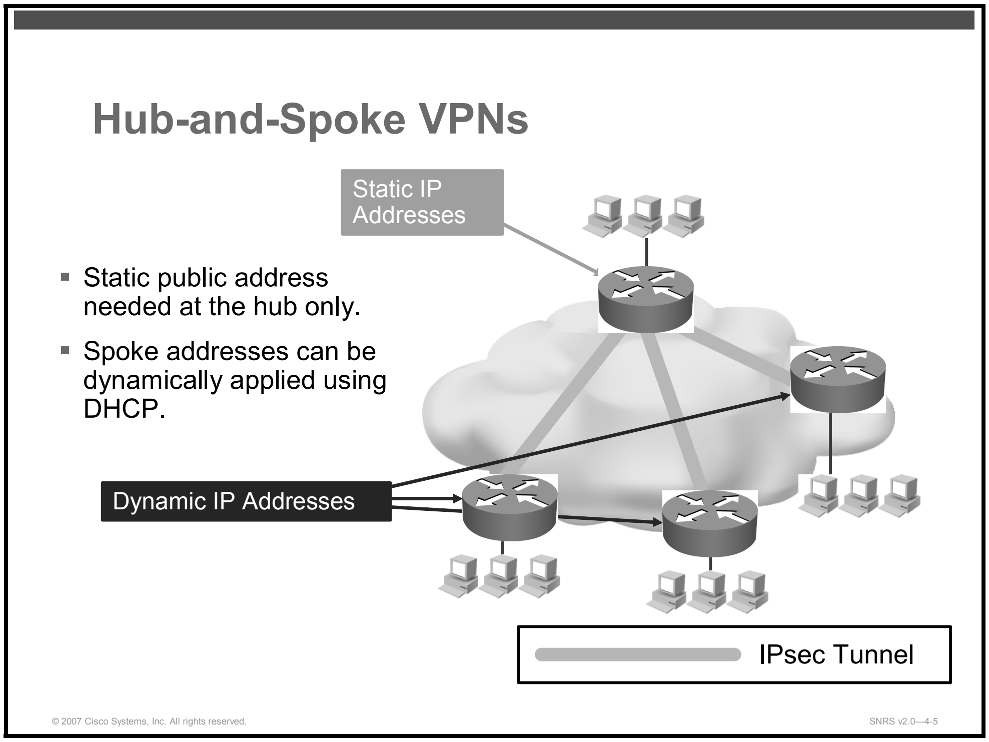
Restrictions

Here are some of the restrictions of a fully meshed VPN topology:

- All sites must have static IP addresses for IPsec peering
- When adding a new site, all other routers have to be reconfigured to add the new site.

Hub-and-Spoke IPsec VPNs

This topic describes hub-and-spoke IPsec VPNs.



Hub-and-Spoke VPNs

- Static public address needed at the hub only.
- Spoke addresses can be dynamically applied using DHCP.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-5

In a hub-and-spoke network configuration, the spoke sites connect with IPsec tunnels to a hub site to establish connectivity to the network. The hub site consists of high-end tunnel aggregation routers servicing multiple IPsec tunnels for a predefined maximum number of spoke locations. Small site routers (spoke sites) typically connect to a set of large site routers (hub sites).

Another benefit of terminating the VPN tunnels at the hub site is that the headend can act as the distribution point for all routing information and connectivity to and from spoke site devices. For resiliency and load distribution, the hub site could be made with multiple headend devices.

When the majority of traffic is targeted to the hub and the core of the network, the hub-and-spoke design is the most suitable configuration. Additional IPsec connections that form partial mesh connections can enable a direct IPsec path if some spoke sites require direct access.

In a hub-and-spoke configuration, the hub typically uses statically assigned public IP addresses, while the spokes can use dynamically assigned IP addresses. In an environment where the spoke sites are also using static public addresses, a partial mesh of IPsec connections can create the VPN using site-to-site configurations.

The main feature for enabling this configuration is the dynamic crypto map, which eases IPsec configuration. Dynamic crypto maps are used in the hub-and-spoke configuration to support the dynamic addresses at the spokes. The peer addresses are not predetermined in the hub configurations and are dynamically assigned IP addresses. The spokes need to authenticate themselves to the hub to establish the IPsec tunnel to the hub. If pre-shared keys are used as the authentication, the hub needs to be configured with a wildcard pre-shared key because spoke IP addresses are not known beforehand. All spokes that know the pre-shared key and whose IP address match the network mask for the wildcard pre-shared key are acceptable for connection to the hub. A dynamic crypto map is essentially a crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured to match the requirements of a remote peer. When using a dynamic crypto map, only the remote peer can initiate the tunnel.

Characteristics

Here are some of the common characteristics of a hub-and-spoke VPN:

- Static IP address at the hub
- Dynamic addressing at the spokes
- Uses dynamic crypto maps
- Uses wildcard IP addresses with the pre-shared keys

Benefits

Here are some of the benefits of a hub-and-spoke VPN topology:

- **Provides support for small sites with small LAN and low-end routers:** Only one IPsec tunnel is needed at the spoke routers.
- **Reduces the hub router configuration size and complexity:** The hub router no longer needs to maintain a separate static crypto map for each of the spoke sites or to maintain a list of IP addresses of the spoke sites, thus simplifying the add, delete, and spoke sites.
- **Only hub needs to have static and global IP address:** All the spoke routers could have a DHCP-based dynamic IP address, with the hub configured with a dynamic crypto map.
- **Very easy to add a new site or router:** No changes to the existing spoke or hub routers are required.

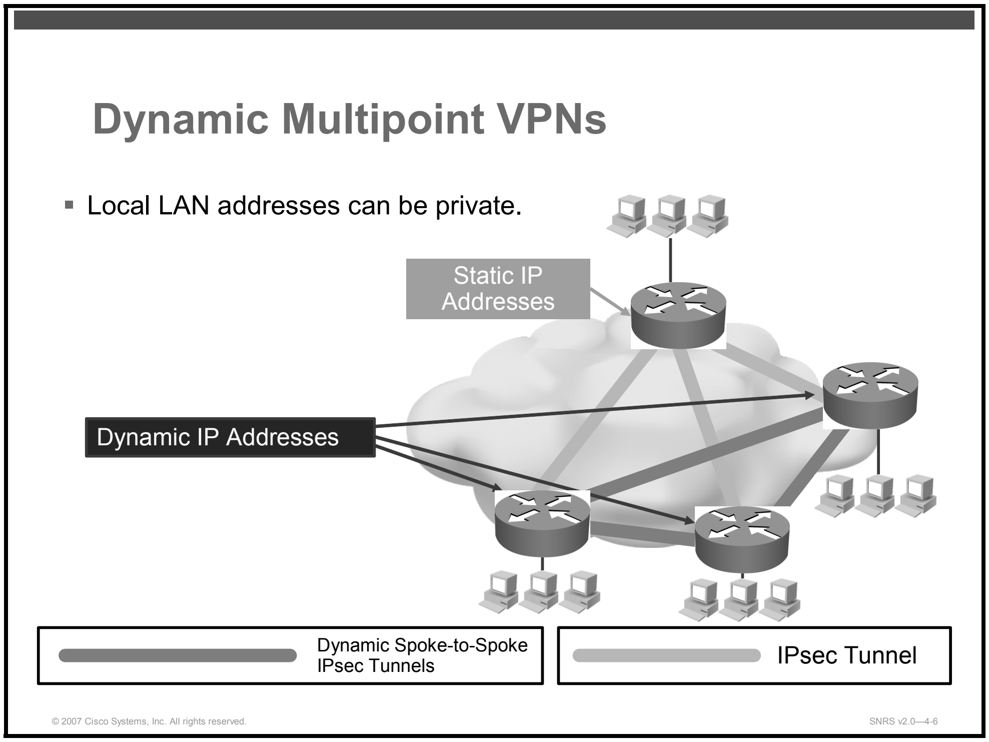
Restrictions

Here are some of the restrictions of a hub-and-spoke VPN topology:

- IPsec performance is aggregated at the hub.
- All spoke packets are decrypted and re-encrypted at the hub.
- When using hub-and-spoke configurations with dynamic crypto maps, the IPsec encryption tunnel must be initiated by the spoke routers.

Dynamic Multipoint VPNs

This topic describes DMVPNs.



Dynamic Multipoint VPNs

- Local LAN addresses can be private.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-6

Some companies may want to interconnect small sites together, while simultaneously connecting to a main site over the Internet. When small sites are interconnected, it is difficult to maintain the configurations for all of the connections. It is also difficult to create, add, and change a large full-mesh network configuration. Because the spokes do have direct access to each other over the Internet, it would be beneficial for the spoke-to-spoke traffic to go directly rather then via a hub site. This would be useful when two spokes are in the same city and the hub is across the country. With the DMVPN IPsec solution, the spoke sites would be able to dynamically establish secure connectivity between them.

DMVPN provides for a combination of static and dynamic on-demand tunnels. The static VPN tunnels are connected to a hub site in a hub-and-spoke fashion. The hub-and-spoke design is the most suitable configuration when the majority of the traffic is targeted to the hub and the core of the network. When some spoke sites require direct access between them, an additional IPsec connection forming a partial-mesh connection will dynamically direct the IPsec path.

DMVPNs use multipoint GRE (mGRE)/Next Hop Resolution Protocol (NHRP) with both IPsec and NHRP to resolve the peer destination address and automatic IPsec encryption initiation.

NHRP also provides the capability for the spoke routers to dynamically learn the exterior physical interface address of the routers in the VPN network.

Note This means that the spoke routers would have enough information to dynamically build an IPsec-mGRE tunnel directly between spoke routers.

This is important because if this spoke-to-spoke data traffic is sent via the hub router, it must be encrypted and decrypted twice, thus increasing the delay and the decryption and encryption of this through traffic and increasing the load on the hub router. To use this feature, the spoke routers need to learn, via the dynamic IP routing protocol running over the IPsec-mGRE tunnel with the hub, the subnetworks that are available behind the other spokes with an IP next hop of the tunnel IP address of the other spoke router.

Each spoke has a permanent IPsec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server where the hub router is the NHRP server. When a spoke needs to send a packet to a destination subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke. After the originating spoke learns the peer address of the target spoke, it can initiate a dynamic IPsec tunnel to the target spoke. The spoke-to-spoke tunnel is built over the mGRE interface. The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets are able to bypass the hub and use the spoke-to-spoke tunnel.

Characteristics

Here are some of the common characteristics of DMVPNs:

- DMVPNs support dynamic IP addresses on the spokes.
- Local LANs can have private addresses.

Benefits

Here are some of the benefits of a DMVPN topology:

- Useful when configuration of spoke-to-spoke traffic is relatively complex to configure and maintain
- Reduces the hub router configuration size and complexity (The hub router no longer needs to maintain a separate static crypto map for each of the spoke sites or to maintain a list of IP addresses of the spoke sites, simplifying the add, delete and spoke sites.)
- Conserves router resources by establishing links on demand and tear down after a preconfigured duration of inactivity
- Split tunneling at the spokes supported
- Creates a constant configuration size on the hub router, regardless of how many spoke routers are added to the VPN network

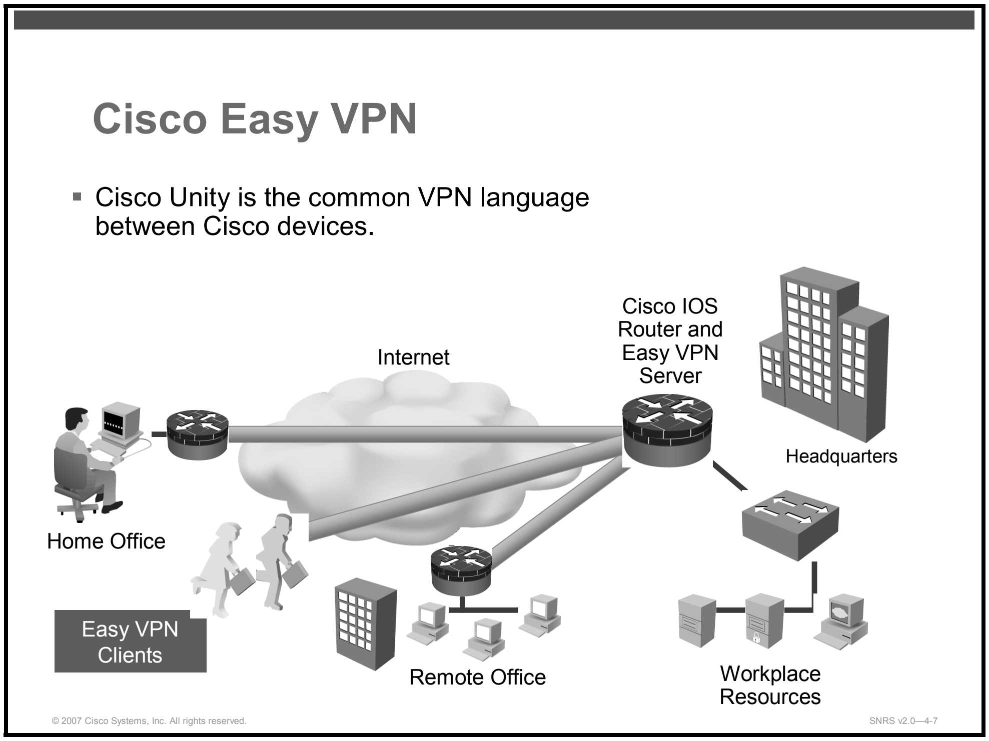
Restrictions

Here are some restrictions of a DMVPN topology:

- The majority of the traffic should be passing the dedicated hub sites to minimize topology changes.
- The initial packets will go through the hub, until the spoke-to-spoke tunnel is established.
- When using hub and spoke with dynamic crypto maps, the IPsec encryption tunnel must be initiated by the spoke routers.

Cisco Easy VPN

This topic describes Cisco Easy VPN.



Cisco Easy VPN

- Cisco Unity is the common VPN language between Cisco devices.

SNRS v2.0—4-7

When deploying VPNs for teleworkers and small branch offices, ease of deployment is increasingly important. Cisco Easy VPN makes it easier than ever to deploy VPNs as part of small and medium businesses or large enterprise networks with Cisco products. Cisco Easy VPN Remote and Cisco Easy VPN Server offer flexibility, scalability, and ease of use for site-to-site and remote-access VPNs.

A router enabled with Cisco Easy VPN Server can terminate VPN tunnels initiated by mobile and remote workers running Cisco VPN Client software on PCs. It also allows remote routers to act as Cisco Easy VPN Remote nodes. Cisco Easy VPN allows the VPN parameters, such as internal IP addresses, internal subnet masks, DHCP server addresses, Microsoft Windows Internet Name Service (WINS) server addresses, and split-tunneling flags, to be pushed from the Cisco Easy VPN Server to the remote device. This server can be a dedicated VPN device, such as a Cisco VPN 3000 Series Concentrator or a Cisco PIX Firewall, or a Cisco IOS router.

Characteristics

The most common characteristic of Cisco Easy VPN is its ease of configuration.

Benefits

Here are some of the benefits of a Cisco Easy VPN topology:

- Centrally stored configurations allow dynamic configuration of end-user policy and require less manual configuration.

- The local VPN configuration is independent of the remote peer IP address. This allows the provider to change equipment and network configurations as needed, with little or no reconfiguration of the end-user equipment.
- Cisco Easy VPN provides for centralized security policy management.
- Cisco Easy VPN enables large-scale deployments with rapid user provisioning.
- Cisco Easy VPN removes the need for end users to install and configure Cisco Easy VPN Remote software on their PCs.

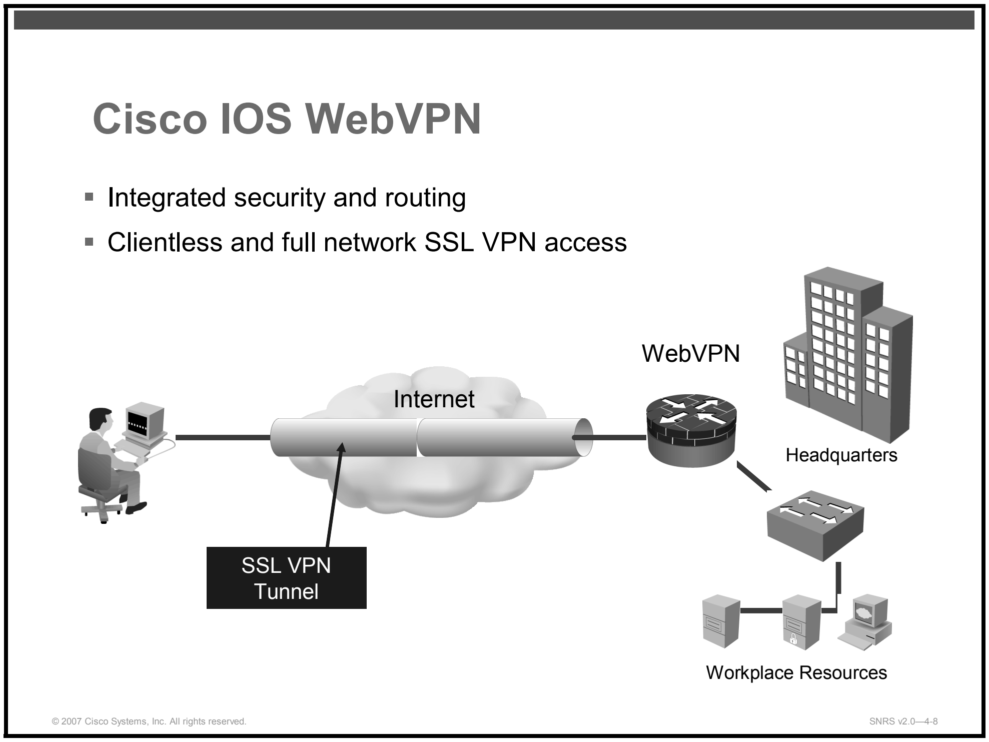
Restrictions

Here are some of the restrictions of Cisco Easy VPN:

- No manual Network Address Translation (NAT) or Port Address Translation (PAT) configuration allowed
  - Cisco Easy VPN Remote automatically creates the appropriate NAT or PAT configuration for the VPN tunnel.
- Only one destination peer supported
  - Cisco Easy VPN Remote supports the configuration of only one destination peer and tunnel connection.
  - If an application requires the creation of multiple VPN tunnels, the IPsec VPN and NAT and PAT parameters on both the remote and server must be manually configured.
- Requires destination servers
  - Cisco Easy VPN Remote requires that the destination peer be a Cisco Easy VPN remote-access server.
- Digital certificates not supported
  - Authentication is supported using pre-shared keys.
  - Extended Authentication (XAUTH) may also be used in addition to pre-shared keys to provide user-level authentication in addition to device level authentication.
- Only Internet Security Association and Key Management Protocol (ISAKMP) policy group 2 supported on IPsec servers
  - The Cisco VPN Client client/server protocol supports only ISAKMP policies that use group 2 (1024-bit Diffie-Hellman [DH]) Internet Key Exchange (IKE) negotiation.
- Some transform sets not supported
- The Cisco Easy VPN Remote feature does not support transform sets that provide encryption without authentication (ESP-DES and ESP-3DES) or transform sets that provide authentication without encryption (ESP-NULL, ESP-SHA-HMAC, and ESP-NULL ESP-MD5-HMAC).
  - The Cisco VPN Client client/server protocol does not support Authentication Header (AH) authentication, but Encapsulation Security Payload (ESP) is supported.

WebVPN

This topic describes WebVPN (Cisco IOS Secure Sockets Layer [SSL] VPN).



Cisco IOS WebVPN

- Integrated security and routing
- Clientless and full network SSL VPN access

SNRS v2.0—4-8

SSL-based VPN, or WebVPN, is an emerging technology that provides remote-access connectivity from almost any Internet-enabled location using a web browser and its native SSL encryption. SSL VPN provides the flexibility to support secure access for all users, regardless of the endpoint host from which they are establishing the connection. If application access requirements are modest, SSL VPN does not require a software client to be preinstalled on the endpoint host. This enables companies to extend their secure enterprise networks to any authorized user by providing remote-access connectivity to corporate resources from any Internet-enabled location.

WebVPN currently delivers two modes of SSL VPN access: clientless and thin client. WebVPNs allow users to access web pages and services (including the ability to access files, send and receive e-mail, and run TCP-based applications) without IPsec VPN client software. WebVPNs are appropriate for user populations that require per-application or per-server access control, or access from nonenterprise-owned desktops.

In many cases, IPsec and WebVPN are complementary, because they solve different problems. This complementary approach allows a single device to address all remote-access user requirements.

Characteristics

The most common characteristic of WebVPN is that it runs on port 443.

Benefits

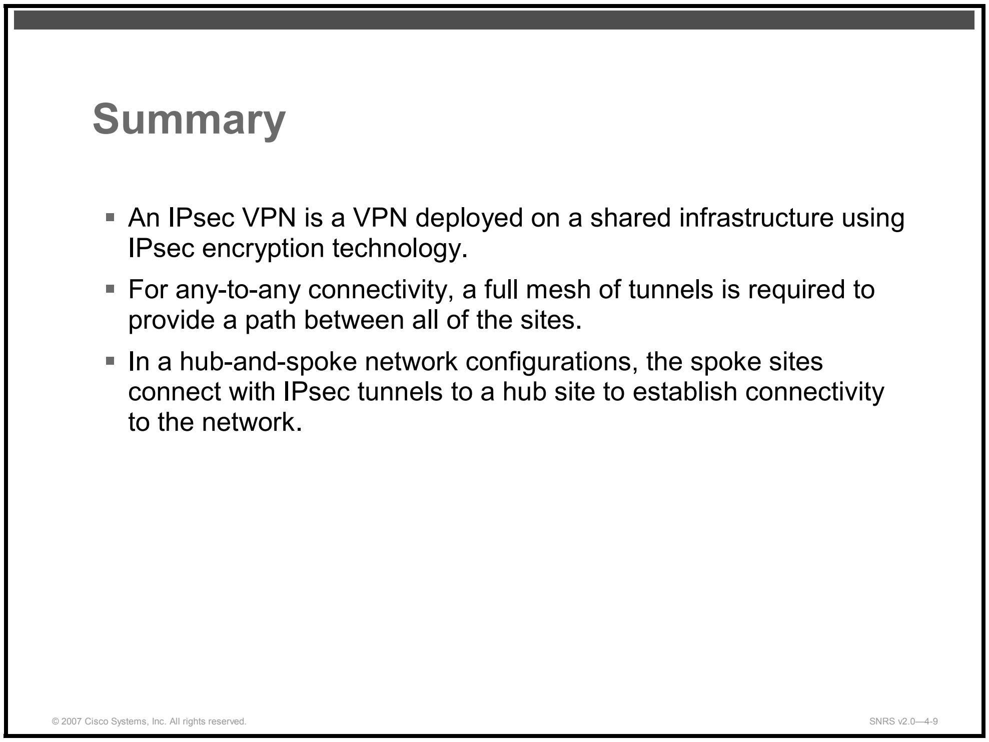
The primary benefit of WebVPN is that it is compatible with DMVPNs, Cisco IOS Firewalls, IPsec, intrusion prevention systems (IPS), Cisco Easy VPN, and NAT.

Restrictions

The primary restriction of WebVPN is that it is currently supported only in software. The router CPU processes the WebVPN connections. The on-board VPN acceleration available in integrated services routers only accelerates IPsec connections.

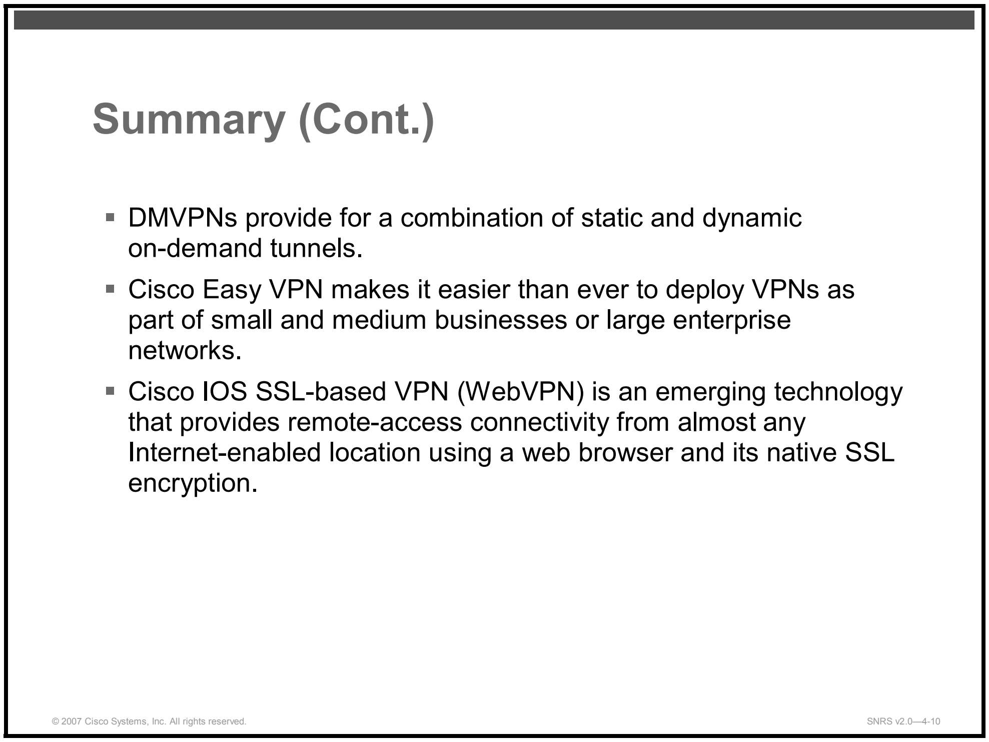
Summary

This topic summarizes the key points that were discussed in this lesson.



Summary

- An IPsec VPN is a VPN deployed on a shared infrastructure using IPsec encryption technology.
- For any-to-any connectivity, a full mesh of tunnels is required to provide a path between all of the sites.
- In a hub-and-spoke network configurations, the spoke sites connect with IPsec tunnels to a hub site to establish connectivity to the network.



Summary (Cont.)

- DMVPNs provide for a combination of static and dynamic on-demand tunnels.
- Cisco Easy VPN makes it easier than ever to deploy VPNs as part of small and medium businesses or large enterprise networks.
- Cisco IOS SSL-based VPN (WebVPN) is an emerging technology that provides remote-access connectivity from almost any Internet-enabled location using a web browser and its native SSL encryption.

References

For additional information, refer to this resource:

- Cisco Systems, Inc. Deploying IPsec Virtual Private Networks.http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns142/networking_solutions_white_paper09186a0080117919.shtml.

Lesson 3

Implementing IPsec VPNs Using Pre-Shared Keys

Overview

IP Security (IPsec) virtual private networks (VPNs) can be configured for various types of authentication. One such method is pre-shared keys. In this case, each IPsec VPN peer shares a common key. This lesson guides you through the process of configuring IPsec site-to-site VPNs using pre-shared keys.

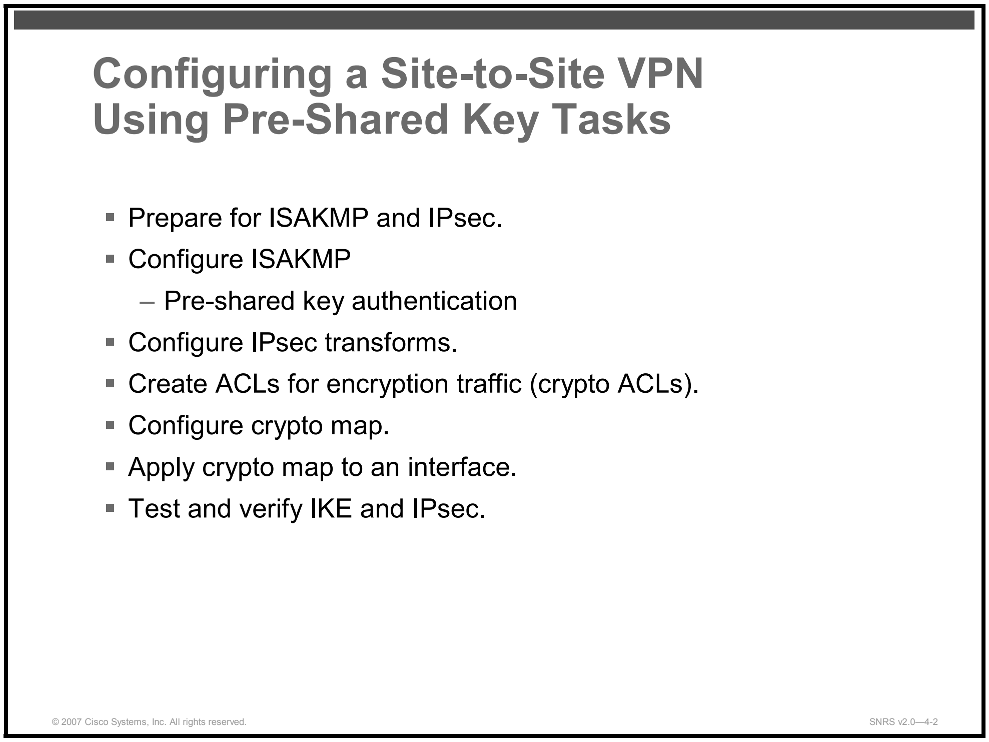
Objectives

Upon completing this lesson, you will be able to configure an IPsec site-to-site VPN using pre-shared keys. This ability includes being able to meet these objectives:

- Describe the tasks involved when configuring a site-to-site IPsec VPN using IKE pre-shared keys for authentication
- Describe how to prepare a network for IPsec configuration
- Determine the IKE policies to be used between IPsec peers
- Determine an IPsec policy to be used between IPsec peers
- Describe how to configure ISAKMP using pre-shared keys
- Describe how to configure IKE with pre-shared keys
- Describe each of the steps used in configuring the IPsec policy
- Describe how to apply crypto maps to interfaces
- Describe the commands used to test and verify an IPsec configuration
- Describe some strategies for troubleshooting IPsec

Configuring IPsec

This topic describes the tasks required to configure an IPsec site-to-site VPN using pre-shared keys.



Configuring a Site-to-Site VPN Using Pre-Shared Key Tasks

- Prepare for ISAKMP and IPsec.
- Configure ISAKMP
  - Pre-shared key authentication
- Configure IPsec transforms.
- Create ACLs for encryption traffic (crypto ACLs).
- Configure crypto map.
- Apply crypto map to an interface.
- Test and verify IKE and IPsec.

SNRS v2.0—4-2

There are several configuration items that must be enabled to implement IPsec on a router.

The major tasks are as follows:

Step 1 Prepare for IPsec.

This task involves checking network connectivity before IPsec is implemented, checking current configuration for IPsec policies if they exist, ensuring that existing access control lists (ACLs) are compatible with IPsec—for example, ensure that Internet Key Exchange (IKE), IPsec (Encapsulating Security Payload [ESP] or Authentication Header [AH]), and Network Address Translation-Traversal (NAT-T) traffic is permitted through the ACL—, identifying the hosts and networks that you wish to protect, determining details about the IPsec peers, and determining the IPsec features that you need.

Step 2 Configure the IKE policy.

This task involves enabling IKE and creating the IKE policies.

Step 3 Configure IPsec transforms and protocol.

This task includes defining the IPsec transform sets.

Step 4 Create ACLs for encryption.

This task includes defining traffic to be encrypted or excluded from encryption.

Step 5 Apply a crypto map to the interface.

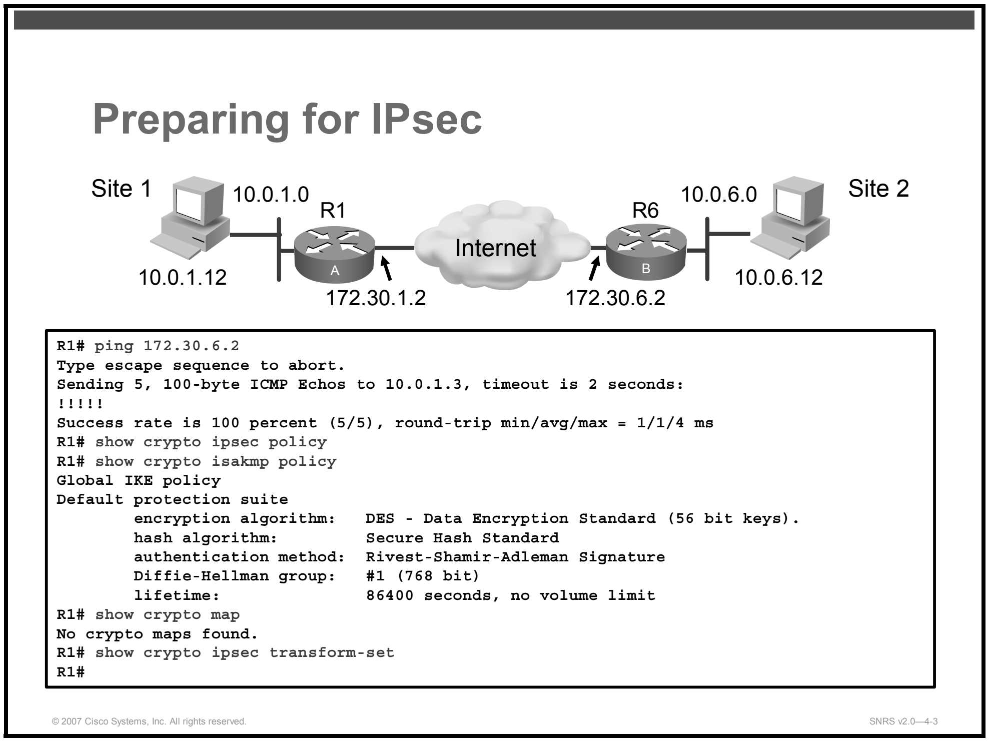
This task applies the IPsec configuration to an interface.

Step 6 Test and verify IKE and IPsec.

In this task, you will use **show**, **debug**, and related commands to test and verify that IKE and IPsec operations are working.

Preparing for IPsec

This topic describes how to prepare a network for IPsec.



Preparing for IPsec

```
R1# ping 172.30.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1# show crypto ipsec policy
R1# show crypto isakmp policy
Global IKE policy
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
R1# show crypto map
No crypto maps found.
R1# show crypto ipsec transform-set
R1#
```

SNRS v2.0—4-3

Preparing for IPsec includes these tasks:

- Checking connectivity without IPsec configured
- Checking for previous IPsec configurations
  - — Checking for any existing Internet Security Association and Key Management Protocol (ISAKMP) or IPsec policies
  - — Checking for any existing crypto maps or transform sets

Checking Connectivity Without IPsec Enabled

Basic connectivity between peers must be checked before you begin configuring IPsec.

The router **ping** command can be used to test basic connectivity between IPsec peers. Although a successful Internet Control Message Protocol (ICMP) echo (ping) will verify basic connectivity between peers, you should ensure that the network works with any other protocols or ports that you want to encrypt, such as Telnet, FTP, or SQL*NET, before beginning IPsec configuration.

After IPsec is activated, basic connectivity troubleshooting can be difficult because the security configuration may mask a more fundamental networking problem.

Checking for Existing IPsec Configurations

You should check the current Cisco router configuration to see if there are any IPsec policies already configured that are useful for, or may interfere with, the IPsec policies that you plan to configure. Previously configured IKE and IPsec policies and details can and should be used, if possible, to save configuration time. However, previously configured IKE and IPsec policies and details can make troubleshooting more difficult if problems arise.

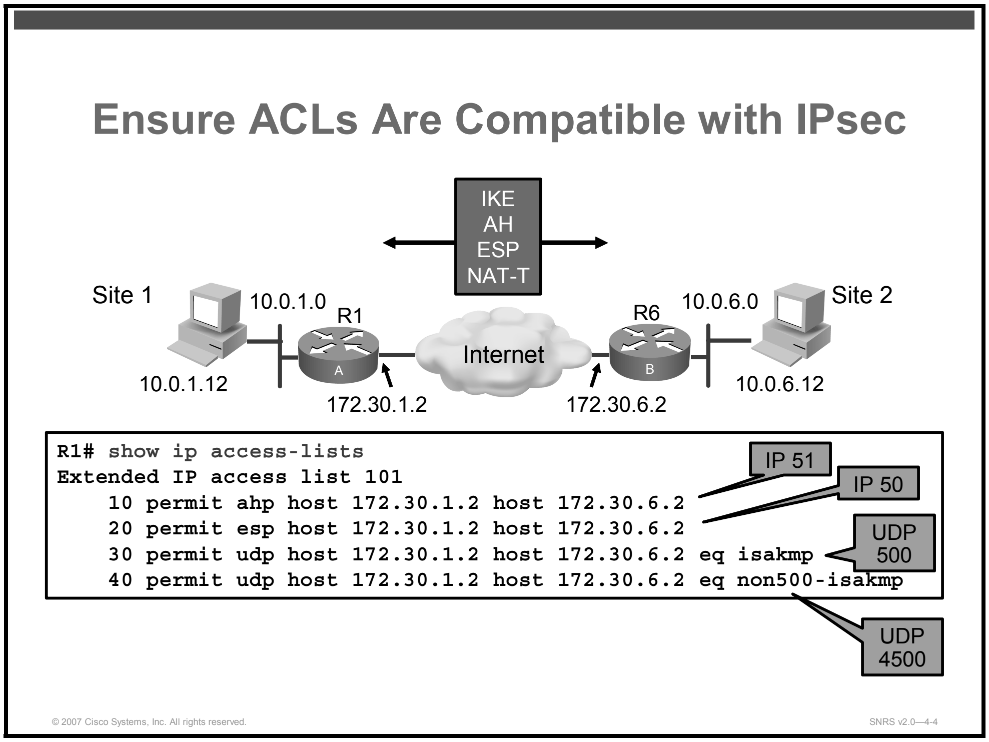
You can see whether any IPsec policies have previously been configured by starting with the **show crypto ipsec policy** and **show crypto isakmp policy** commands**,** as shown in the figure, to examine existing policies. The default ISAKMP protection suite seen in the figure is available for use without modification.

The **show crypto map** command shown in the figure is useful for viewing any previously configured crypto maps. Previously configured maps can and should be used to save configuration time. However, previously configured crypto maps can interfere with the IPsec policy that you are trying to configure.

You can also use the **show crypto ipsec transform-set** command to view previously configured transform sets. Previously configured transforms can be used to save configuration time.

Ensuring ACLs Are Compatible with IPsec

This section describes how to ensure that the existing ACLs on perimeter routers do not block IPsec traffic.



Ensure ACLs Are Compatible with IPsec

```
R1# show ip access-lists
Extended IP access list 101
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```

SNRS v2.0—4-4

You will need to ensure that existing ACLs on perimeter routers, the Cisco Adaptive Security Appliance (ASA) or Cisco PIX Firewall, or other routers do not block IPsec traffic. Perimeter routers typically implement a restrictive security policy with ACLs, where only specific traffic is permitted and all other traffic is denied. Such a restrictive policy blocks IPsec traffic, so you need to add specific permit statements to the ACL to allow IPsec traffic.

Ensure that your ACLs are configured so that ISAKMP, ESP, AH, and NAT-T traffic is not blocked at interfaces used by IPsec. ISAKMP uses User Datagram Protocol (UDP) port 500. ESP is assigned IP protocol 50; AH is assigned IP protocol 51; and NAT-T uses UDP 4500. You might need to these statements to router ACLs to explicitly permit this traffic.

You may need to add the ACL statements to the perimeter router by completing the following steps:

Step 1 Examine the current ACL configuration at the perimeter router and determine whether it will block IPsec traffic.

```
router# show ip access-lists
```

Step 2 Add ACL entries to permit IPsec traffic.

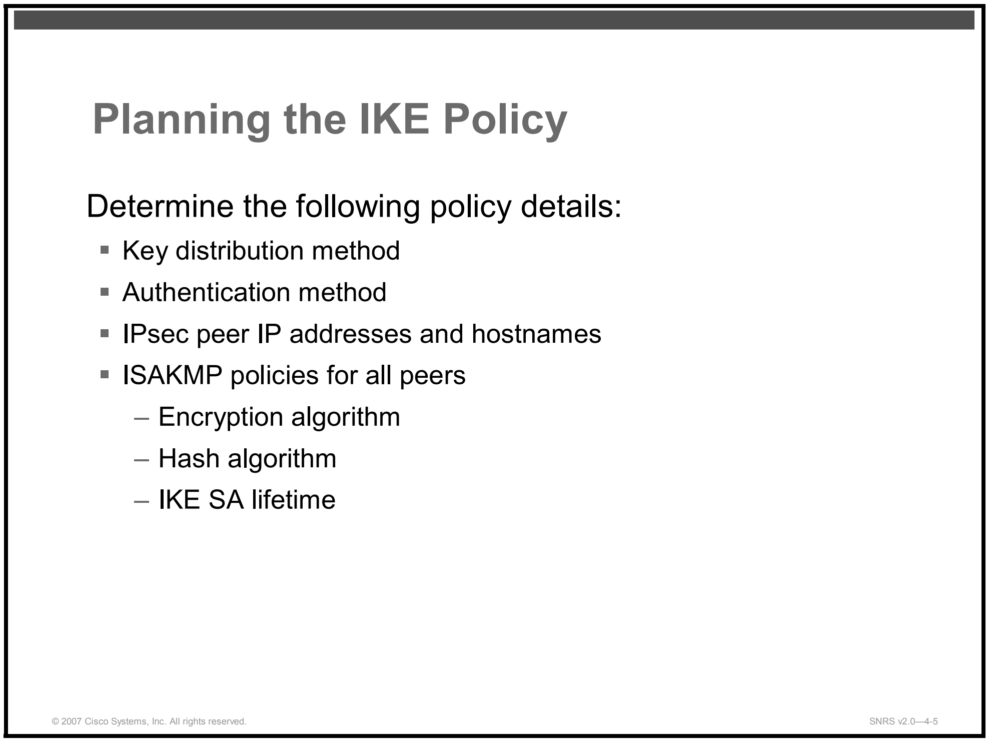
```
R1(config)# ip access-list extended 101
R1(config-ext-nacl)# permit ahp host 172.30.1.2 host
172.30.6.2
```

```
R1(config-ext-nacl)# permit esp host 172.30.1.2 host
172.30.6.2
R1(config-ext-nacl)# permit udp host 172.30.1.2 host
172.30.6.2 eq isakmkp
R1(config-ext-nacl)# permit udp host 172.30.1.2 host
172.30.6.2 eq 4500
```

The figure shows an example of an ACL that is compatible with IPsec and should not block any protocols needed to set up the IPsec tunnels.

Planning the IKE Policy

This topic describes how to determine the IKE policies between IPsec peers.



Planning the IKE Policy

Determine the following policy details:

- Key distribution method
- Authentication method
- IPsec peer IP addresses and hostnames
- ISAKMP policies for all peers
  - Encryption algorithm
  - Hash algorithm
  - IKE SA lifetime

SNRS v2.0—4-5

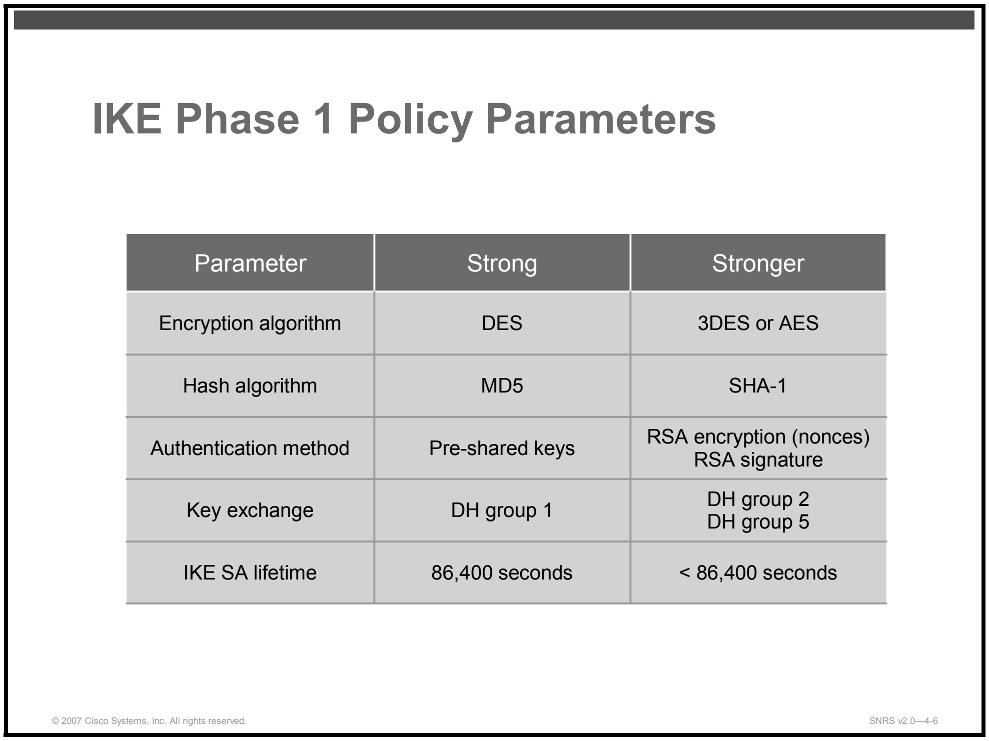
You should determine the ISAKMP (IKE Phase 1) policy details that you want to use and then configure those policy details. Having a detailed IKE policy plan lessens the chances of improper configuration.

Here are some planning steps:

- **Determine the key distribution method:** Determine the key distribution method based on the numbers and locations of IPsec peers. For a small network, you may wish to manually distribute keys. For larger networks, you may wish to use a certificate authority (CA) server to support scalability of IPsec peers. You must then configure ISAKMP to support the selected key distribution method.
- **Determine the authentication method:** Choose the authentication method based on the key distribution method. Cisco IOS Software supports either pre-shared keys, Rivest, Shamir, and Adleman (RSA)-encrypted nonces, or RSA signatures to authenticate IPsec peers. This lesson focuses on using pre-shared keys.
- **Identify the IP addresses and hostnames of IPsec peers:** Determine the details of all of the IPsec peers that will use ISAKMP and pre-shared keys for establishing SAs. You will use this information to configure IKE.

- **Determine ISAKMP policies for peers:** An ISAKMP policy defines a combination or suite of security parameters to be used during the ISAKMP negotiation. Each ISAKMP negotiation begins by each peer agreeing on a common (shared) ISAKMP policy. The ISAKMP policy suites must be determined in advance of configuration. You must then configure IKE to support the policy details that you determined. Some ISAKMP policy details include these:
  - Encryption algorithm
  - Hash algorithm
  - IKE SA lifetime

The goal of this planning step is to gather the precise data that you will need in later steps to minimize misconfiguration.



IKE Phase 1 Policy Parameters

Parameter	Strong	Stronger
Encryption algorithm	DES	3DES or AES
Hash algorithm	MD5	SHA-1
Authentication method	Pre-shared keys	RSA encryption (nonces) RSA signature
Key exchange	DH group 1	DH group 2 DH group 5
IKE SA lifetime	86,400 seconds	< 86,400 seconds

An IKE policy defines a combination of security parameters used during the IKE negotiation. A group of policies makes up a protection suite of multiple policies that enable IPsec peers to establish IKE sessions and establish SAs with a minimal configuration. The figure shows an example of possible combinations of IKE parameters into either a strong or stronger policy suite.

Create IKE Policies for a Purpose

IKE negotiations must be protected, so each IKE negotiation begins by each peer agreeing on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations.

After the two peers agree upon a policy, an SA established at each peer identifies the security parameters of the policy. These SAs apply to all subsequent IKE traffic during the negotiation.

You can create multiple, prioritized policies at each peer to ensure that at least one policy will match the policy of a remote peer.

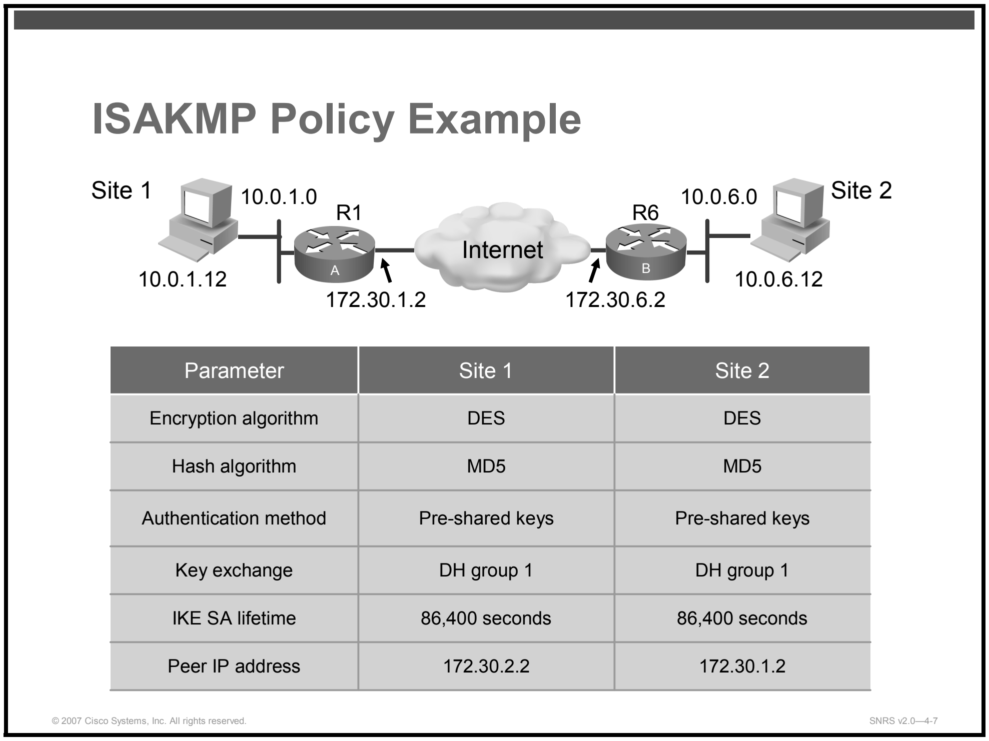
Define IKE Policy Parameters

You can select specific values for each IKE parameter per the IKE standard. You choose one value over another based on the security level that you desire and the type of IPsec peer to which you will connect.

There are five parameters to define in each IKE policy, as outlined in the figure and in the table. The figure shows the relative strength of each parameter; the table shows the default values.

Parameter	Accepted Values	Keyword	Default
Message encryption algorithm	Data Encryption Standard (DES) Triple DES (3DES) Advanced Encryption Standard (AES) 128, 192, or 256 bits	**des** **3des** **aes**	DES
Message integrity (hash) algorithm	Secure Hash Algorithm-1 (SHA-1), Hashed Method Authentication Code (HMAC) (HMAC variant), Message Digest 5 (MD5) (HMAC variant)	**sha** **md5**	SHA-1
Peer authentication method	Pre-shared keys, RSA encrypted nonces, RSA signatures	**pre-share** **rsa-encr** **rsa-sig**	RSA signatures
Key exchange parameters (Diffie-Hellman [DH] group identifier)	768-bit DH, 1024-bit DH, 1536-bit DH	**1** **2** **5**	768-bit DH group 1
ISAKMP-established SA lifetime	Specify any number of seconds	**—**	86,400 seconds (1 day)

You can select specific values for each ISAKMP parameter per the ISAKMP standard. You choose one value over another based on the security level that you desire and the type of IPsec peer to which you will connect.



ISAKMP Policy Example

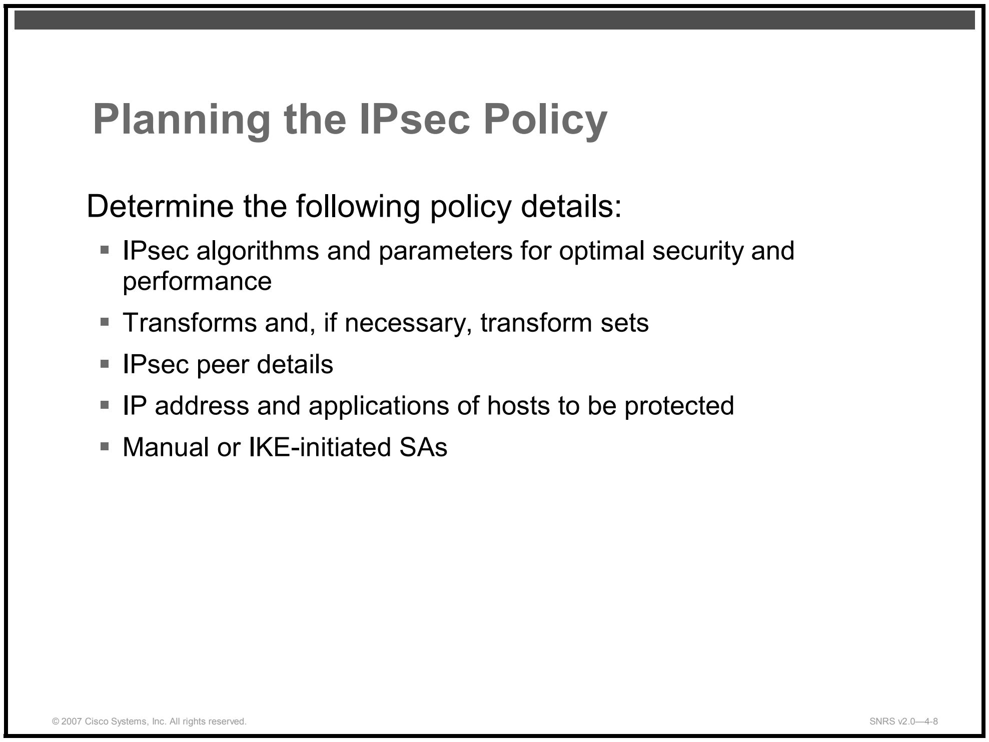
Parameter	Site 1	Site 2
Encryption algorithm	DES	DES
Hash algorithm	MD5	MD5
Authentication method	Pre-shared keys	Pre-shared keys
Key exchange	DH group 1	DH group 1
IKE SA lifetime	86,400 seconds	86,400 seconds
Peer IP address	172.30.2.2	172.30.1.2

SNRS v2.0—4-7

You should determine IKE policy details for each peer before configuring IKE. The figure shows a summary of IKE policy details that will be configured in examples and in labs for this lesson. The authentication method of pre-shared keys is covered in this lesson.

Planning the IPsec Policy

This topic describes how to determine an IPsec policy to be used between IPsec peers.



Planning the IPsec Policy

Determine the following policy details:

- IPsec algorithms and parameters for optimal security and performance
- Transforms and, if necessary, transform sets
- IPsec peer details
- IP address and applications of hosts to be protected
- Manual or IKE-initiated SAs

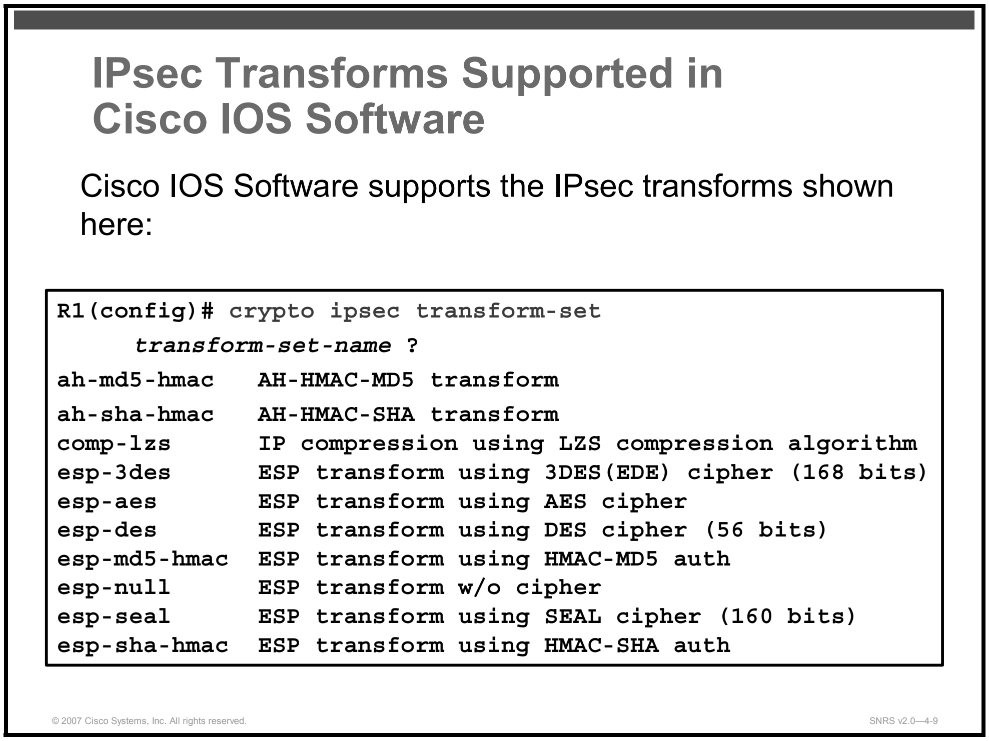
SNRS v2.0—4-8

An IPsec policy defines a combination of IPsec parameters used during the IPsec negotiation. Planning for IPsec (IKE Phase 2) is another important step that you should complete before actually configuring IPsec on a Cisco router.

Policy details to determine at this stage include these:

- **Select IPsec algorithms and parameters for optimal security and performance:** Determine what type of IPsec security to use when securing interesting traffic. Some IPsec algorithms require you to make tradeoffs between high performance and stronger security. Some algorithms have import and export restrictions that may delay or prevent the implementation of your network.
- **Select transforms and, if necessary, transform sets:** Use the IPsec algorithms and parameters previously decided upon to help select IPsec transforms, transform sets, and modes of operation (tunnel mode or transport mode).
- **Identify IPsec peer details:** Identify the IP addresses and hostnames of all IPsec peers to which you will connect.
- **Determine IP address and applications of hosts to be protected:** Decide which host IP addresses and applications should be protected at the local peer and the remote peer.
- **Select manual or IKE-initiated SAs:** Choose whether SAs are manually established or are established via IKE.

The goal of this planning step is to gather the precise data that you will need in later steps to minimize misconfiguration.



IPsec Transforms Supported in Cisco IOS Software

Cisco IOS Software supports the IPsec transforms shown here:

```
R1(config)# crypto ipsec transform-set
     transform-set-name ?
ah-md5-hmac   AH-HMAC-MD5 transform
ah-sha-hmac   AH-HMAC-SHA transform
comp-lzs      IP compression using LZS compression algorithm
esp-3des      ESP transform using 3DES(EDE) cipher (168 bits)
esp-aes       ESP transform using AES cipher
esp-des       ESP transform using DES cipher (56 bits)
esp-md5-hmac  ESP transform using HMAC-MD5 auth
esp-null      ESP transform w/o cipher
esp-seal      ESP transform using SEAL cipher (160 bits)
esp-sha-hmac  ESP transform using HMAC-SHA auth
```

SNRS v2.0—4-9

Cisco IOS Software supports the IPsec transforms shown in the tables.

Transform	Description
ah-md5-hmac	AH-HMAC-MD5 transform
ah-sha-hmac	AH-HMAC-SHA transform

AH is rarely used because authentication is now available with the **esp-sha-hmac** and **esp-md5-hmac** transforms. AH is also not compatible with Network Address Translation (NAT) or Port Address Translation (PAT).

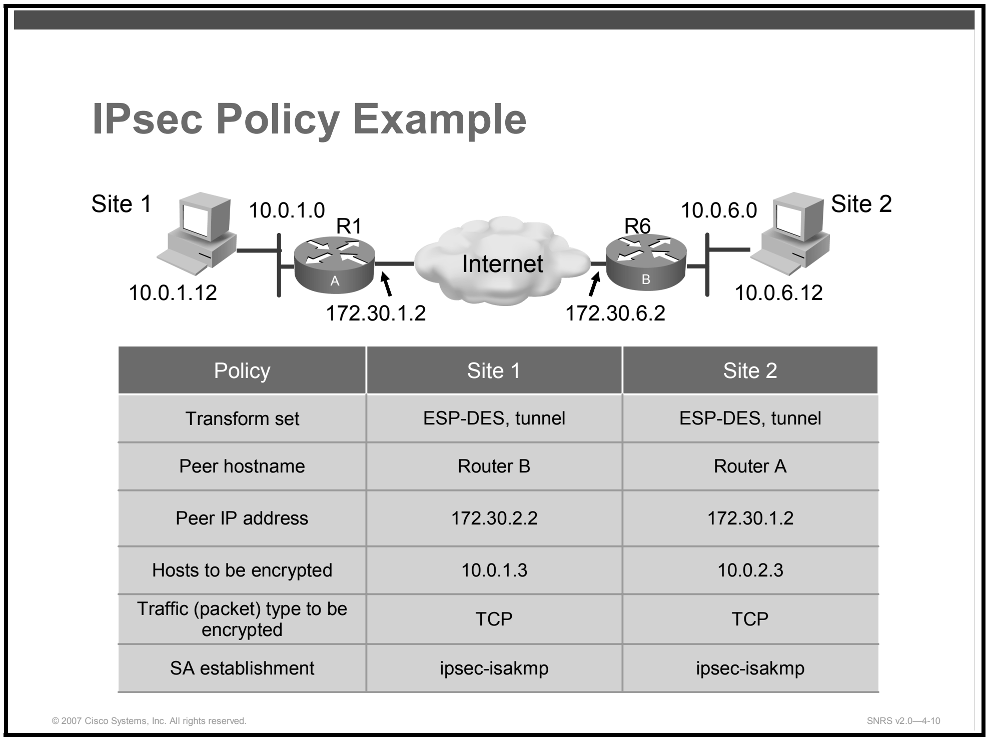
Transform	Description
esp-des	ESP transform using DES cipher (56 bits)
esp-3des	ESP transform using 3DES-EDE cipher (168 bits)
esp-aes	ESP transform using AES cipher (128, 192, or 256 bits)
esp-md5-hmac	ESP transform with HMAC-MD5 authentication used with an **esp-des or esp-3des t**ransform to provide additional integrity of ESP packet
esp-sha-hmac	ESP transform with HMAC-SHA authentication used with an **esp-des** or **esp-3des t**ransform to provide additional integrity of ESP packet
esp-null	ESP transform without a cipher; may be used in combination with **esp-md5-hmac** or **esp-sha-hmac t**ransform if one wants ESP authentication with no encryption

Caution Never use the **esp-null** transform in a production environment because it does not protect data flows.

Examples of acceptable transforms that can be combined into sets are shown in the table.

Transform Type	Allowed Transform Combinations
AH transform (Choose up to one)	■ **ah-md5-hmac:** AH with MD5 (HMAC variant) authentication algorithm ■ **ah-sha-hmac:** AH with SHA (HMAC variant) authentication algorithm
ESP encryption transform (Choose up to one)	■ **esp-des:** ESP with 56-bit DES encryption algorithm ■ **esp-3des:** ESP with 168-bit DES encryption algorithm (3DES) ■ **esp-aes:** ESP transform using AES cipher (128, 192, or 256 bits) ■ **esp-null:** Null encryption algorithm
ESP authentication transform (Choose up to one)	■ **esp-md5-hmac:** ESP with MD5 (HMAC variant) authentication algorithm ■ **esp-sha-hmac:** ESP with SHA (HMAC variant) authentication algorithm
IP compression transform	■ **comp-lzs:** IP compression with the Lempel-Ziv-Stac (LZS) algorithm

The Cisco IOS command parser prevents you from entering invalid combinations; for example, after you specify an AH transform, it does not allow you to specify another AH transform for the current transform set.



IPsec Policy Example

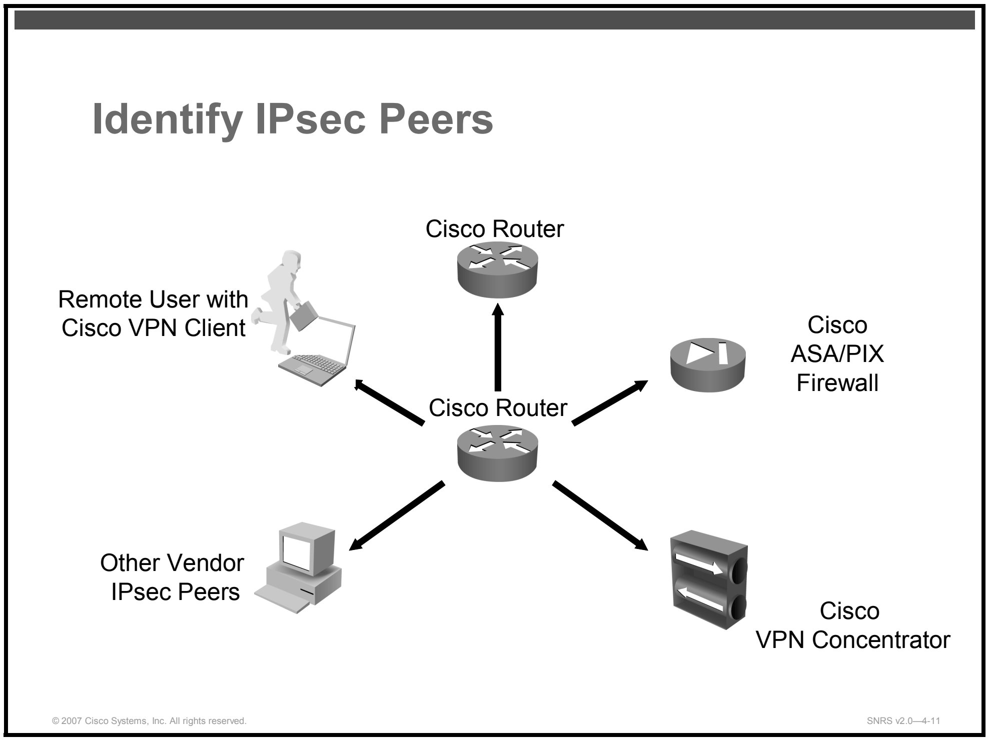
Policy	Site 1	Site 2
Transform set	ESP-DES, tunnel	ESP-DES, tunnel
Peer hostname	Router B	Router A
Peer IP address	172.30.2.2	172.30.1.2
Hosts to be encrypted	10.0.1.3	10.0.2.3
Traffic (packet) type to be encrypted	TCP	TCP
SA establishment	ipsec-isakmp	ipsec-isakmp

SNRS v2.0—4-10

Determining network design details includes defining a more detailed IPsec policy for protecting traffic. You can then use the detailed policy to help select IPsec transform sets and modes of operation. Your IPsec policy should answer the following questions:

- What protections are required or are acceptable for the protected traffic?
- Which IPsec transforms or transform sets should be used?
- What are the peer IPsec endpoints for the traffic?
- What traffic should or should not be protected?
- Which router interfaces are involved in protecting internal networks and external networks?
- How are SAs set up (manually or IKE-negotiated), and how often should the SAs be renegotiated?

The figure shows a summary of IPsec encryption policy details that will be configured in examples in this lesson. Example policy specifies that TCP traffic between the hosts should be encrypted by IPsec using DES.



Identify IPsec Peers

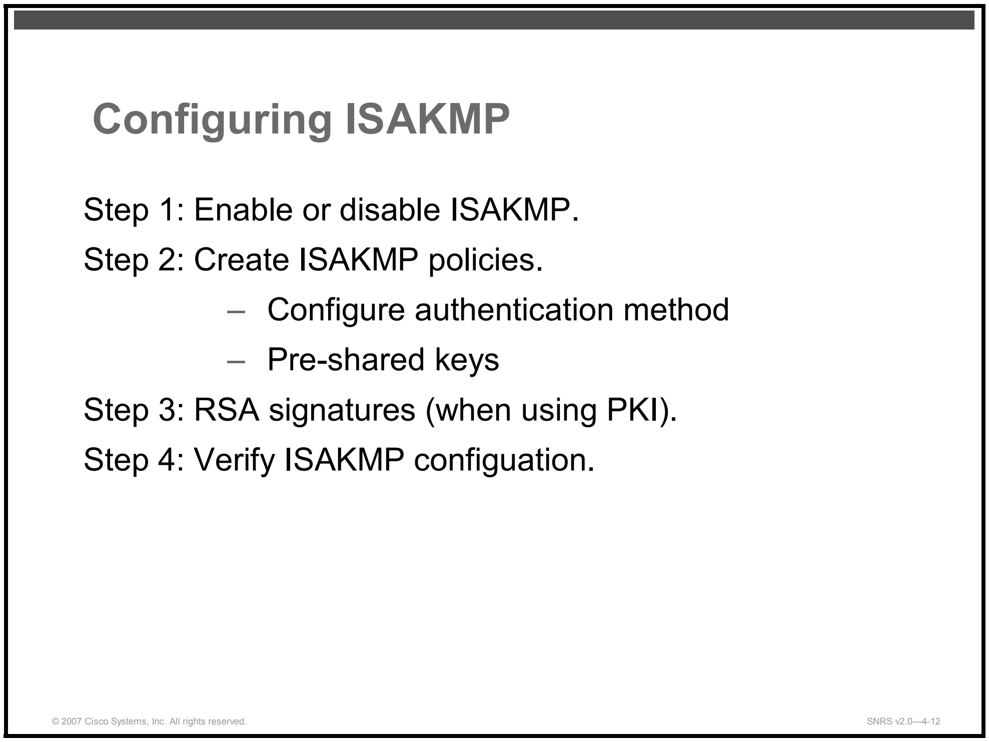
SNRS v2.0—4-11

An important part of determining the IPsec policy is to identify the IPsec peer that the Cisco router will communicate with. The peer must support IPsec as specified in the RFCs as supported by Cisco IOS Software. Many different types of peers are possible. Before configuration, identify all the potential peers and their VPN capabilities. Possible peers include, but are not limited to, the following:

- Other Cisco routers
- Cisco ASA or Cisco PIX Firewall
- Cisco VPN Client
- Cisco VPN concentrator
- IPsec products from other vendors that conform to IPsec RFCs

Configuring ISAKMP

This topic describes how to configure ISAKMP using pre-shared keys.



Configuring ISAKMP

Step 1: Enable or disable ISAKMP.

Step 2: Create ISAKMP policies.

- Configure authentication method
- Pre-shared keys

Step 3: RSA signatures (when using PKI).

Step 4: Verify ISAKMP configuation.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-12

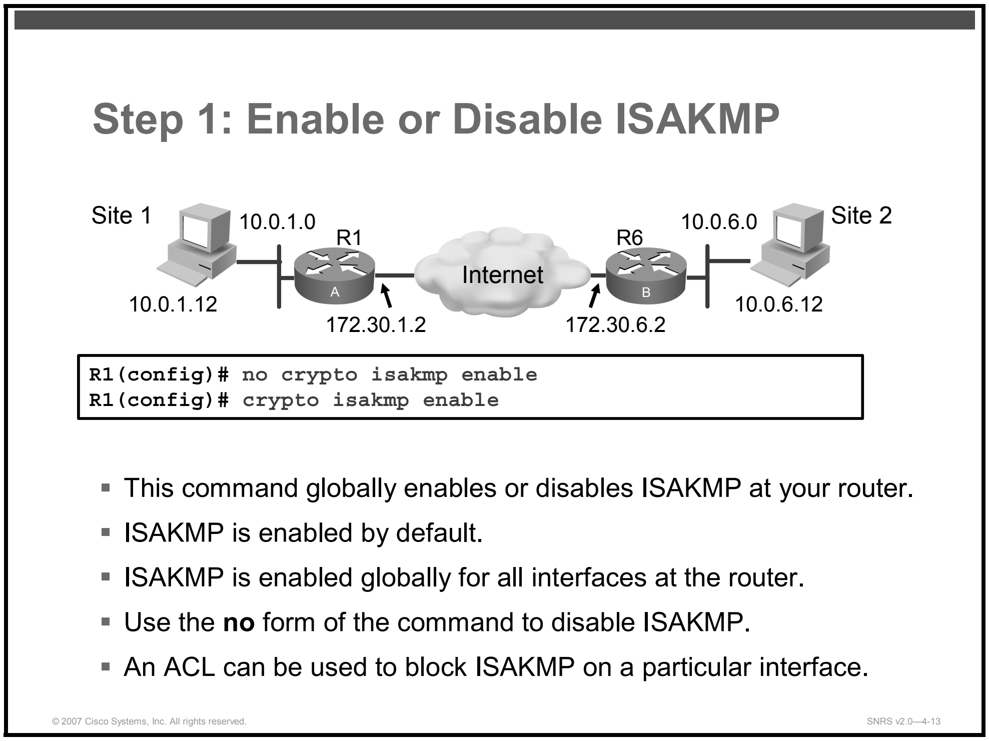
IKE automatically negotiates IPsec SAs and enables IPsec secure communications without costly manual reconfigurations. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec SAs . Multiple IKE policies can be defined between two IPsec peers; however, there must be at least one matching IKE policy between them to establish the IPsec tunnels.

Configuring IKE policies involves the following tasks:

- Enabling or disabling ISAKMP globally
- Creating ISAKMP policies
- Configuring pre-shared keys
- Verifying ISAKMP configuration

Enable or Disable ISAKMP

This section describes how to enable or disable ISAKMP.



Step 1: Enable or Disable ISAKMP

```
R1(config)# no crypto isakmp enable
R1(config)# crypto isakmp enable
```

- This command globally enables or disables ISAKMP at your router.
- ISAKMP is enabled by default.
- ISAKMP is enabled globally for all interfaces at the router.
- Use the **no** form of the command to disable ISAKMP.
- An ACL can be used to block ISAKMP on a particular interface.

SNRS v2.0—4-13

IKE is enabled by default. IKE does not have to be enabled for individual interfaces, but it is enabled globally for all interfaces at the router. You may choose to block ISAKMP access on interfaces not used for IPsec to prevent possible denial of service (DoS) attacks by using an ACL statement that blocks UDP port 500 on the interfaces.

If you do not want IKE to be used with your IPsec implementation, you can disable it at all IPsec peers.

If you disable IKE, you will have to make these concessions at the peers:

- You must manually specify all the IPsec SAs in the crypto maps at all peers.
- The IPsec SAs of the peers will never time out for a given IPsec session.
- During IPsec sessions between the peers, the encryption keys will never change.
- Antireplay services will not be available between the peers.
- CA support cannot be used.

To enable IKE, use the following command:

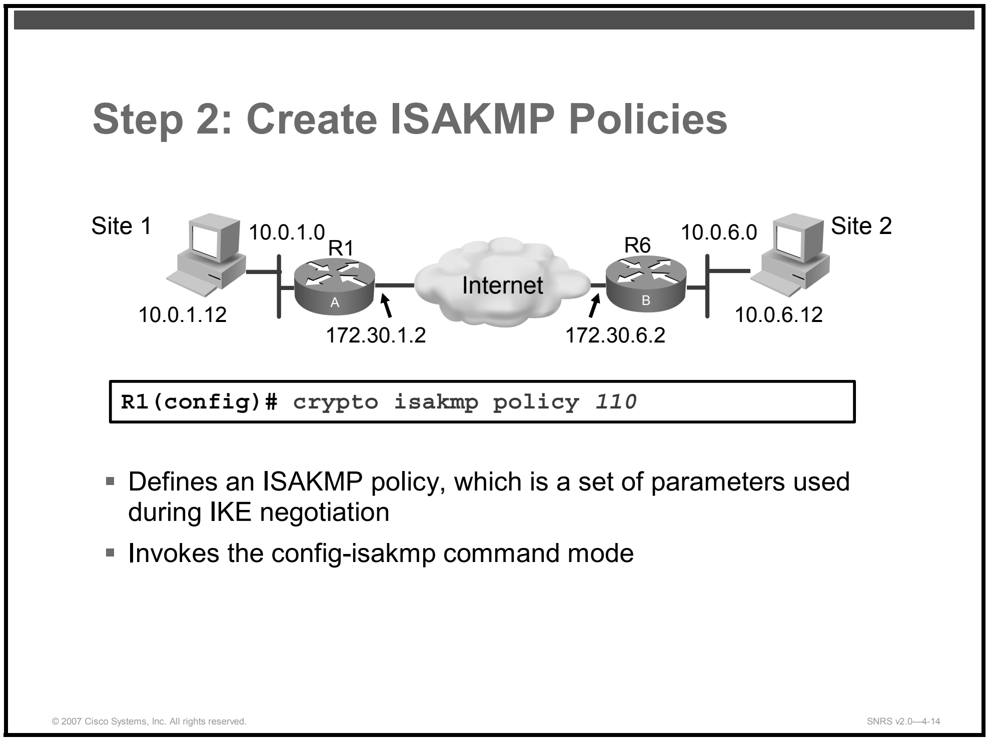
```
Router(config)# crypto isakmp enable
```

To disable IKE, use the following command:

```
Router(config)# no crypto isakmp enable
```

Create ISAKMP Policies

This section describes how to create ISAKMP policies.



Step 2: Create ISAKMP Policies

```
R1(config)# crypto isakmp policy 110
```

- Defines an ISAKMP policy, which is a set of parameters used during IKE negotiation
- Invokes the config-isakmp command mode

You must create ISAKMP policies at each peer. An ISAKMP policy defines a combination of security parameters to be used during the IKE negotiation.

Why Do You Need to Create These Policies?

IKE negotiations must be protected, so each IKE negotiation begins by the agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can create multiple, prioritized policies at each peer to ensure that at least one policy will match the policy of a remote peer.

Parameters Defined in a Policy

There are five parameters to define in each IKE policy, as shown in the table.

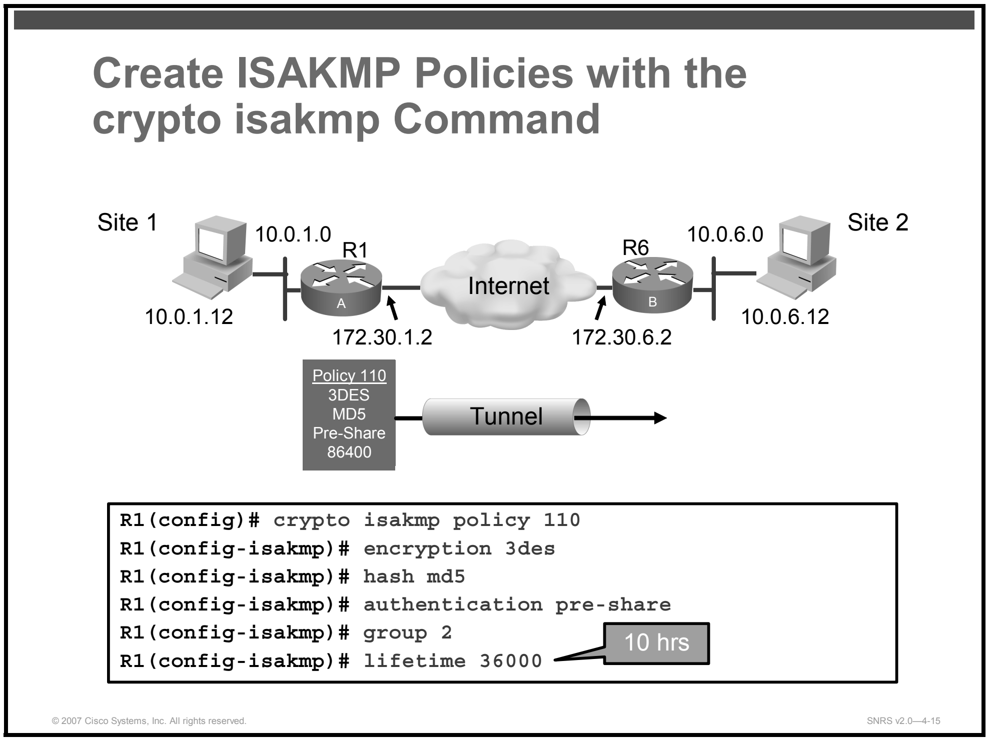
Parameter	Keyword	Description	Default Value
Encryption algorithm	**Des**	56-bit DES with cipher block chaining (CBC)	56-bit DES with CBC
	3des		168-bit DES
	aes		128-bit AES
	aes 192 **aes 256**		192-bit AES 256-bit AES
Hash algorithm	**sha**	SHA-1 (HMAC variant)	SHA-1
	md5	MD5 (HMAC variant)	
Authentication method	**rsa-sig** **rsa-encr**	RSA signatures; RSA encrypted nonces	RSA signatures
	pre-share	pre-shared keys	
DH group identifier	**1**	768-bit DH group 1	768-bit DH group 1
	2	1024-bit DH group 2	
	5	1536-bit DH group 5	
Lifetime of the SA	60 seconds to 86,400 seconds		86,400 seconds (1 day)

These parameters apply to the IKE negotiations when the IKE SA is established.

You can create multiple IKE policies, each with a different combination of parameter values. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

You can configure multiple policies on each peer; however, at least one of these policies must contain exactly the same encryption, hash, authentication, and DH parameter values as one of the policies on the remote peer.

If you do not configure any policies, your router will use the default policy, which is always set to the lowest priority and which contains the default value of each parameter.



Create ISAKMP Policies with the crypto isakmp Command

```
R1(config)# crypto isakmp policy 110
R1(config-isakmp)# encryption 3des
R1(config-isakmp)# hash md5
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 2
R1(config-isakmp)# lifetime 36000
```

SNRS v2.0—4-15

To define an IKE policy, follow these steps:

Step 1 Identify the policy to create and enter the **ISAKMP configuration** command mode. (Each policy is uniquely identified by the priority number that you assign.)

```
router(config)# crypto isakmp policy priority
```

Syntax Description

priority	Uniquely identifies the IKE policy and assigns a priority to the policy; use an integer from 1 to 10,000, with 1 being the highest priority and 10,000 the lowest.

Use this command to specify the parameters to be used during an IKE negotiation. (These parameters are used to create the IKE SA.)

This command invokes the ISAKMP policy configuration (**config-isakmp**) command mode. While in the ISAKMP policy configuration command mode, some of the commands for which you can specify parameters are as follows:

- **encryption** (IKE policy); default = 56-bit DES with CBC
- **hash** (IKE policy); default = SHA-1
- **authentication**; default = RSA signatures
- **group** (IKE policy); default = 768-bit DH
- **lifetime** (IKE policy); default = 86,400 seconds (1 day)

If you do not specify any given parameter, the default value will be used for that parameter.

To exit the **config-isakmp** command mode, enter **exit**.

You can configure multiple IKE policies on each peer participating in IPSec. When the IKE negotiation begins, it tries to find a common policy configured on both peers, starting with the highest priority policies as specified on the remote peer.

Step 2 Specify the encryption algorithm.

```
router(config-isakmp)# encryption {des | 3des | aes | aes 192
| aes 256}
```

Syntax Description

des	56-bit DES with CBC as the encryption algorithm
3des	168-bit DES (3DES) as the encryption algorithm
aes	128-bit AES as the encryption algorithm
aes 192	192-bit AES as the encryption algorithm
aes 256	256-bit AES as the encryption algorithm

Use this command to specify the encryption algorithm to be used in an IKE policy.

If a user enters an IKE encryption method that the hardware does not support, a warning message will be displayed immediately after the **encryption** command is entered.

Step 3 Specify the hash algorithm.

```
router(config-isakmp)# hash {sha | md5}
```

Syntax Description

sha	Specifies SHA-1 (HMAC variant) as the hash algorithm
md5	Specifies MD5 (HMAC variant) as the hash algorithm

Use this command to specify the hash algorithm to be used in an IKE policy.

Step 4 Specify the authentication method.

```
Router(config-isakmp)# authentication {rsa-sig | rsa-encr |
pre-share}
```

Syntax Description

rsa-sig	Specifies RSA signatures as the authentication method This method is not supported in IP version 6 (IPv6).
rsa-encr	Specifies RSA encrypted nonces as the authentication method This method is not supported in IPv6.
pre-share	Specifies pre-shared keys as the authentication method

Use this command to specify the authentication method to be used in an IKE policy.

If you specify RSA signatures, you must configure your peer routers to obtain certificates from a CA.

If you specify RSA-encrypted nonces, you must ensure that each peer has the public keys of the other peer.

If you specify pre-shared keys, you must also separately configure these pre-shared keys.

Step 5 Specify the DH group identifier.

```
router(config-isakmp)# group {1 | 2 | 5}
```

Syntax Description

1	Specifies the 768-bit DH group
2	Specifies the 1024-bit DH group
5	Specifies the 1536-bit DH group

Use this command to specify the DH group to be used in an IKE policy.

Step 6 Specify the lifetime of the SA.

```
router(config-isakmp)# lifetime seconds
```

Syntax Description

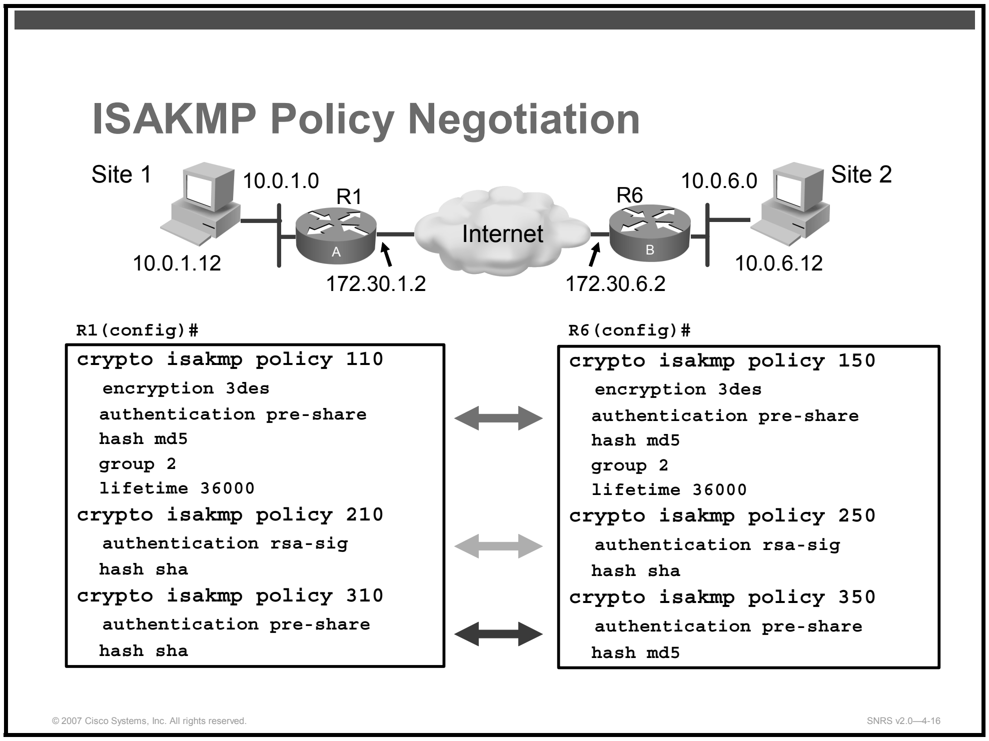
seconds	Specifies how many seconds each SA should exist before expiring. Use an integer from 60 to 86,400 seconds, which is the default value.

Use this command to specify how long an IKE SA exists before expiring.

When IKE begins negotiations, the first thing it does is agree upon the security parameters for its own session. The agreed-upon parameters are then referenced by an SA at each peer. The SA is retained by each peer until the lifetime of the SA expires. Before an SA expires, it can be reused by subsequent IKE negotiations, which can save time when setting up new IPsec SAs. New IPsec SAs are negotiated before current IPsec SAs expire.

So, to save setup time for IPsec, configure a longer IKE SA lifetime. However, shorter lifetimes limit the exposure to attackers of this SA. The longer an SA is used, the more encrypted traffic can be gathered by an attacker and possibly used in an attack.

Note that when your local peer initiates an IKE negotiation between itself and a remote peer, an IKE policy can be selected only if the lifetime of the remote peer policy is shorter than or equal to the lifetime of the local peer policy. Then, if the lifetimes are not equal, the shorter lifetime will be selected. In other words, if the policy lifetimes of the two peers are not the same, the lifetime of the initiating peer must be longer and the lifetime of the responding peer must be shorter, and the shorter lifetime will be used.



ISAKMP Policy Negotiation

```
R1(config)#
crypto isakmp policy 110
  encryption 3des
  authentication pre-share
  hash md5
  group 2
  lifetime 36000
crypto isakmp policy 210
  authentication rsa-sig
  hash sha
crypto isakmp policy 310
  authentication pre-share
  hash sha
```

```
R6(config)#
crypto isakmp policy 150
  encryption 3des
  authentication pre-share
  hash md5
  group 2
  lifetime 36000
crypto isakmp policy 250
  authentication rsa-sig
  hash sha
crypto isakmp policy 350
  authentication pre-share
  hash md5
```

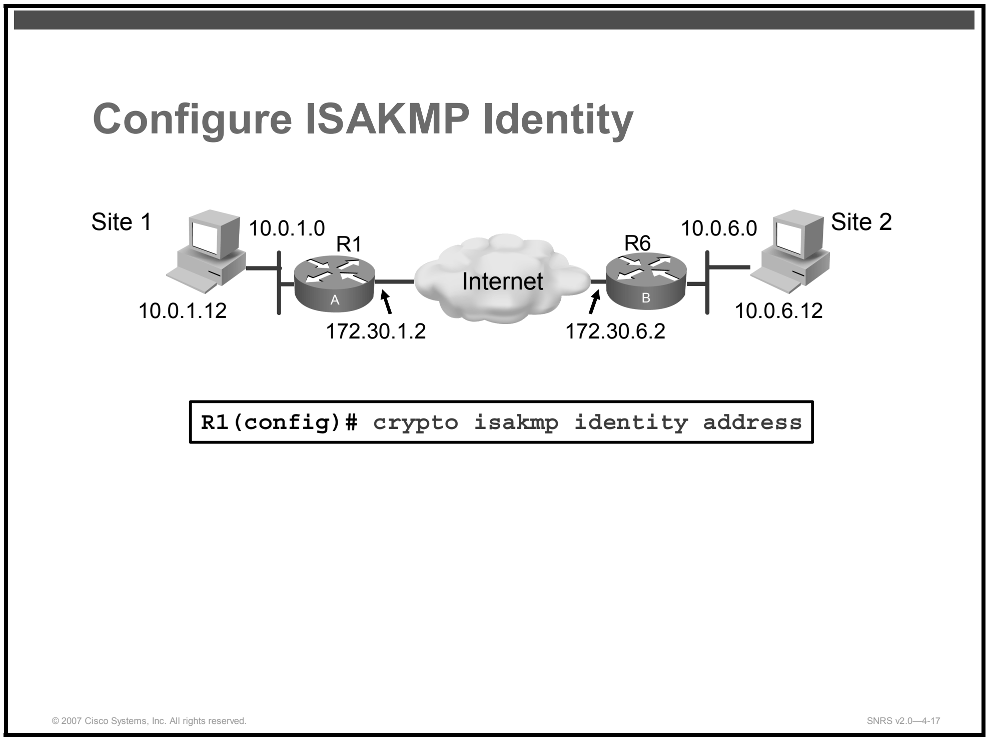
© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-16

ISAKMP peers negotiate acceptable ISAKMP policies before agreeing upon the SA to be used for IPsec.

When the ISAKMP negotiation begins in IKE Phase 1 main mode, ISAKMP looks for an ISAKMP policy that is the same on both peers. The peer that initiates the negotiation sends all its policies to the remote peer, and the remote peer tries to find a match with its policies. The remote peer looks for a match by comparing its own highest-priority policy against the policies received from its other peer in its ISAKMP policy suite. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and DH parameter values, and when the remote peer policy specifies a lifetime less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime from the remote peer policy is used.) Assign the most secure policy the lowest priority number so that the most secure policy will find a match before any less secure policies are configured.

If no acceptable match is found, ISAKMP refuses negotiation and IPsec is not established. If a match is found, ISAKMP completes the main mode negotiation, and IPsec SAs are created during IKE Phase 2 quick mode.



Configure ISAKMP Identity

```
R1(config)# crypto isakmp identity address
```

SNRS v2.0—4-17

You should set the ISAKMP identity for each peer that uses pre-shared keys in an IKE policy.

When two peers use IKE to establish IPsec SAs, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IP address, depending on how you have set the ISAKMP identity of the router.

By default, the ISAKMP identity of a peer is the IP address of the peer. If appropriate, you could change the identity to be the peer hostname instead. As a general rule, either all peers should use their IP addresses or all peers should use their hostnames.

Caution Make sure that you set the identities of all peers the same way.

If some peers use their hostnames and some peers use their IP addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a Domain Name System (DNS) lookup is unable to resolve the identity.

To set the ISAKMP identity of a peer, follow these steps:

Step 1 At the local peer, specify the peer ISAKMP identity by IP address or by hostname.

```
Router(config)# crypto isakmp identity {address | dn |
hostname}
```

Syntax Description

address	Sets the ISAKMP identity to the IP address of the interface that is used to communicate to the remote peer during IKE negotiations
dn	Uses the distinguished name of the router certificate for the identity

hostname	Sets the ISAKMP identity to the hostname concatenated with the domain name (for example, myhost.example.com)

Use this command to specify an ISAKMP identity either by IP address or by hostname.

The **address** keyword is typically used when there is only one interface (and, therefore, only one IP address) that will be used by the peer for IKE negotiations, and the IP address is known.

The **hostname** keyword should be used if there is more than one interface on the peer that might be used for IKE negotiations, or if the IP address of the interface is unknown (such as with dynamically assigned IP addresses).

Step 2 At all remote peers, if the local peer ISAKMP identity was specified using a hostname, map the hostname of the peer to its IP address or addresses at all the remote peers. (This step might be unnecessary if the hostname or address is already mapped in a DNS server.)

```
Router(config)# ip host [vrf vrf-name] {name | tmodem-telephone-
number} [tcp-port-number] address1 [address2...address8]
```

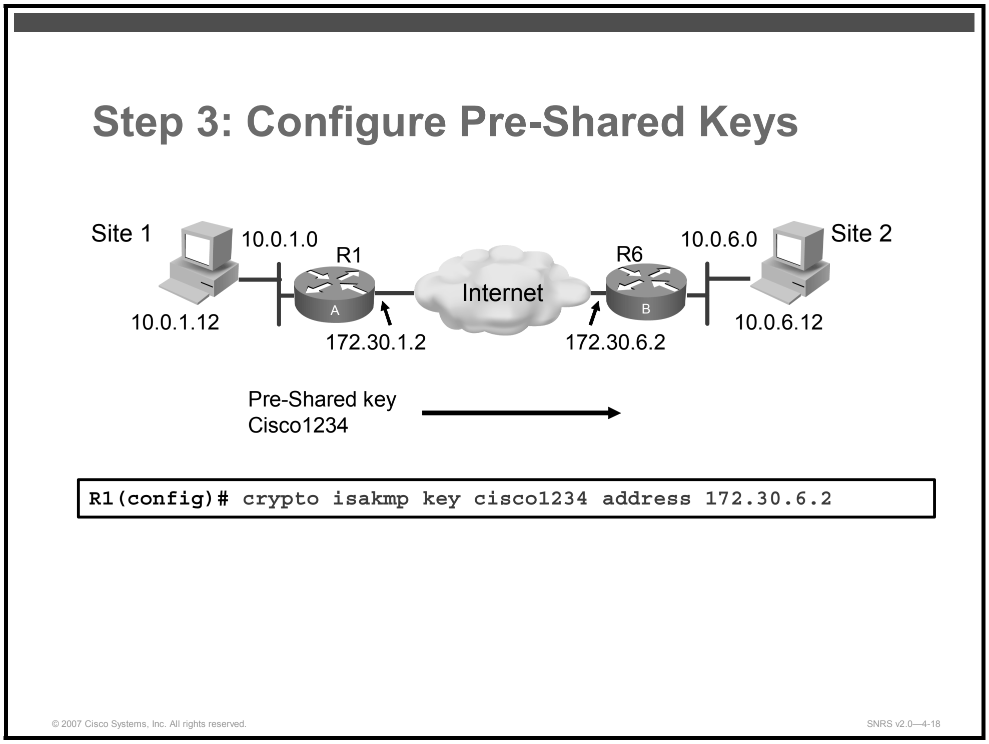
Syntax Description

vrf *vrf-name*	(Optional) Defines a VPN routing and forwarding instance (VRF) table The *vrf-name* argument specifies a name for the VRF table.
name	Name of the host The first character can be either a letter or a number. If you use a number, the types of operations that you can perform are limited.
t*modem-telephone-number*	Modem telephone number that is mapped to the IP host address for use in Cisco modem user interface mode You must enter the letter "t" before the telephone number.
tcp-port-number	(Optional) TCP port number to connect to when using the defined hostname in conjunction with an EXEC mode connect or **telnet** command The default is Telnet (port 23).
address1	Associated IP host address
address2...address8	(Optional) Additional associated IP addresses You can bind up to eight addresses to a hostname.

Remember to repeat these tasks at each peer that uses pre-shared keys in an IKE policy.

Configure Pre-Shared Keys

This topic describes how to configure IKE with pre-shared keys.



Step 3: Configure Pre-Shared Keys

```
R1(config)# crypto isakmp key cisco1234 address 172.30.6.2
```

SNRS v2.0—4-18

The pre-shared key is used to identify and authenticate the IPsec tunnel. The key can be any arbitrary alphanumeric key up to 128 characters long—the key is case-sensitive and must be entered identically on both routers. The configuration in the figure uses a unique pre-shared key that is tied to a specific IP address.

To configure pre-shared keys, complete these steps at each peer that uses pre-shared keys in an IKE policy:

Step 1 Specify the pre-shared keys at each peer. Note that a given pre-shared key is shared between two peers. At a given peer, you could specify the same key to share with multiple remote peers; however, a more secure approach is to specify different keys to share between different pairs of peers.

```
Router(config)# crypto isakmp key enc-type-digit {keystring}
{address peer-address [mask] | ipv6 {ipv6-address/ipv6-prefix}
| hostname hostname} [no-xauth]
```

Syntax Description

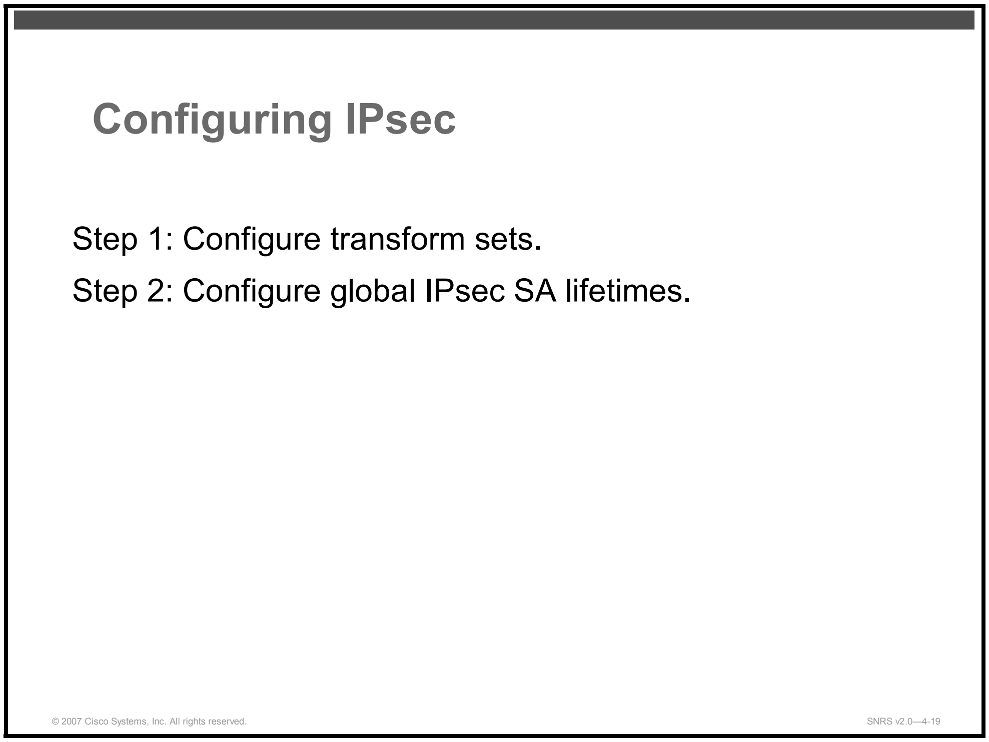
enc-type-digit	Specifies whether the password to be used is encrypted or unencrypted: ■ **0:** Specifies that an unencrypted password follows ■ **6:** Specifies that an encrypted password follows
keystring	Specifies the pre-shared key Use any combination of alphanumeric characters up to 128 bytes. This pre-shared key must be identical at both peers.

`address`	Use this keyword if the remote peer ISAKMP identity was set with its IP or IPv6 address. The *peer-address* argument specifies the IP or IPv6 address of the remote peer.
`peer-address`	Specifies the IP address of the remote peer
`mask`	(Optional) Specifies the subnet address of the remote peer (The argument can be used only if the remote peer ISAKMP identity was set with its IP address.)
`ipv6`	Specifies that an IPv6 address of a remote peer will be used
`ipv6-address`	IPv6 address of the remote peer This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal format using 16-bit values between colons.
`ipv6-prefix`	IPv6 prefix of the remote peer
`hostname` *`hostname`*	Fully qualified domain name (FQDN) of the peer The **hostname** keyword and *hostname* argument are not supported by IPv6.
`no-xauth`	(Optional) Use this keyword if router-to-router IPsec is on the same crypto map as a VPN client to Cisco IOS IPsec. This keyword prevents the router from prompting the peer for Extended Authentication (XAUTH) information (username and password).

Note If the local peer specified its ISAKMP identity with an address, use the **address** keyword in this step; otherwise use the **hostname** keyword in this step.

Configuring IPsec Policies

This topic describes each of the steps used in configuring the IPsec policy.



Configuring IPsec

Step 1: Configure transform sets.

Step 2: Configure global IPsec SA lifetimes.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-19

There are several tasks required to configure IPsec policies on a router.

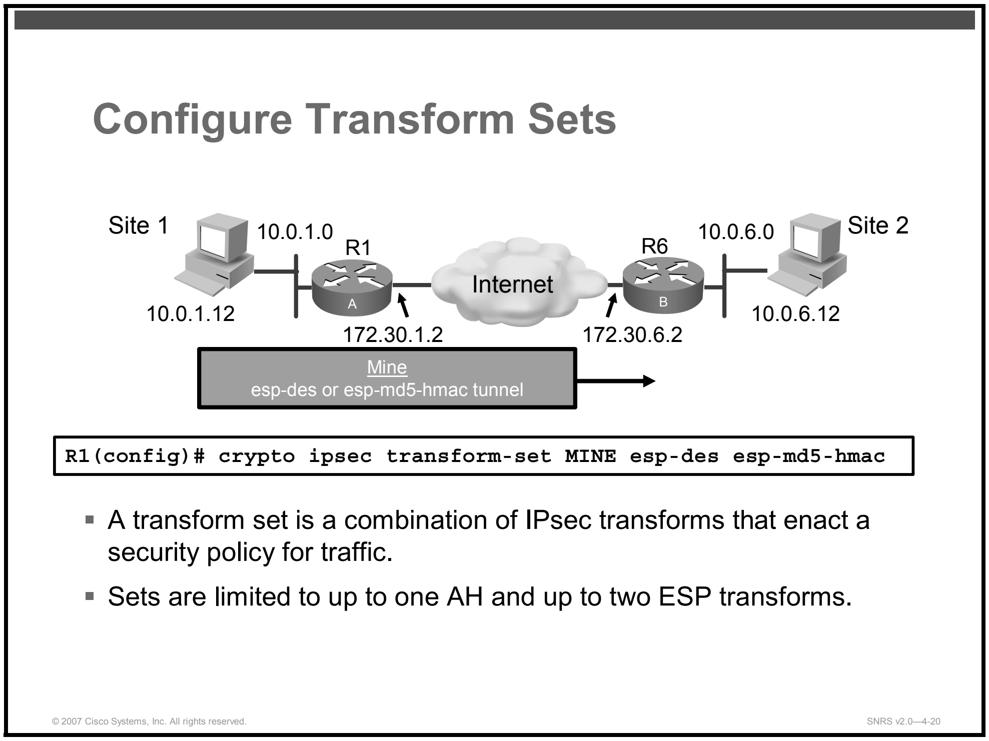
The general tasks and commands used to configure IPsec encryption on Cisco routers are summarized in these steps. Subsequent topics of this lesson discuss each configuration step in detail.

Step 1 Configure transform set suites with the **crypto ipsec transform-set** command.

Step 2 Configure global IPsec SA lifetimes with the **crypto ipsec security-association lifetime** command.

Configuring Transform Sets

This section describes how to define a transform set.



Configure Transform Sets

```
R1(config)# crypto ipsec transform-set MINE esp-des esp-md5-hmac
```

- A transform set is a combination of IPsec transforms that enact a security policy for traffic.
- Sets are limited to up to one AH and up to two ESP transforms.

SNRS v2.0—4-20

A transform set represents a certain combination of security protocols, algorithms, and other settings to apply to IPsec-protected traffic. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry will be used in the IPsec SA negotiation to protect the data flows specified by the ACL of that crypto map entry.

During IPsec SA negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and to be applied to the protected traffic as part of the IPsec SAs of both peers.

With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.

A transform set specifies one or two IPsec security protocols (either AH, ESP, or both) and specifies which algorithms to use with the selected security protocol.

To define a transform set, you specify one to four transforms—each transform represents an IPsec security protocol (AH or ESP) plus the algorithm you want to use. When the particular transform set is used during negotiations for IPsec SAs, the entire transform set (the combination of protocols, algorithms, and other settings) must match a transform set at the remote peer.

In a transform set, you can specify the AH protocol, the ESP protocol, or both. If you specify an ESP protocol in a transform set, you can specify just an ESP encryption transform set or both an ESP encryption transform set and an ESP authentication transform set.

Note If you change a transform set definition, the change is applied only to crypto map entries that reference the transform set. The change will not be applied to existing SAs, but will be used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command.

The ESP with SEAL Transform Set

There are three limitations on the use of the **esp-seal** transform set:

- The **esp-seal** transform set can be used only if no crypto accelerators are present. This limitation is present because no current crypto accelerators implement the Software Encryption Algorithm (SEAL) encryption transform set, and if a crypto accelerator is present, it will handle all IPsec connections that are negotiated with IKE. If a crypto accelerator is present, the Cisco IOS Software will allow the transform set to be configured, but it will warn that it will not be used as long as the crypto accelerator is enabled.
- The **esp-seal** transform set can be used only in conjunction with an authentication transform set, namely one of these: **esp-md5-hmac, esp-sha-hmac**, **ah-md5-hmac**, or **ah-sha-hmac**. This limitation is present because SEAL encryption is especially weak when it comes to protecting against modifications of the encrypted packet. Therefore, to prevent such a weakness, an authentication transform set is required. (Authentication transform sets are designed to foil such attacks.) If you attempt to configure an IPsec transform set using SEAL but without an authentication transform set, an error is generated, and the transform set is rejected.
- The **esp-seal** transform set cannot be used with a manually keyed crypto map. This limitation is present because such a configuration would reuse the same key stream for each reboot, which would compromise security. Because of the security issue, such a configuration is prohibited. If you attempt to configure a manually keyed crypto map with a SEAL-based transform set, an error is generated, and the transform set is rejected.

Selecting Appropriate Transform Sets

The following tips may help you select transform sets that are appropriate for your situation:

- If you want to provide data confidentiality, include an ESP encryption transform set.
- If you want to ensure data authentication for the outer IP header and the data, include an AH transform set. (Some consider the benefits of outer IP header data integrity to be debatable.)
- If you use an ESP encryption transform set, also consider including an ESP authentication transform set or an AH transform set to provide authentication services for the transform set.
- If you want data authentication (either using ESP or AH), you can choose from the MD5 or SHA (HMAC keyed hash variants) authentication algorithms. The SHA algorithm is generally considered stronger than MD5 but is slower.
- Note that some transform sets might not be supported by the IPsec peer.

- In cases where you need to specify an encryption transform set but do not actually encrypt packets, you can use the **esp-null** transform.

Suggested transform set combinations are as follows:

- **esp-3des** and **esp-sha-hmac**
- **esp-aes** and **esp-md5-hmac**

Crypto Transform Configuration Mode

After you issue the **crypto ipsec transform-set** command, you are put into the crypto transform configuration mode. While in this mode, you can change the mode to tunnel or transport. Enter **exit** to return to global configuration mode after you have made any changes.

Follow these steps to define transform sets:

Step 1 Define a transform set.

```
router(config)# crypto ipsec transform-set transform-set-name
transform1 [transform2] [transform3] [transform4]
```

Syntax Description

transform-set-name	Name of the transform set to create (or modify)
transform1 *transform2* *transform3* *transform4*	Type of transform set You may specify up to four transforms: one AH, one ESP encryption, one ESP authentication, and one compression. These transforms define the IPsec security protocols and algorithms.

Before a transform set can be included in a crypto map entry, it must be defined using this command. The table shows the allowed transform combinations.

Allowed Transform Combinations

Transform Type	**Transform**	**Description**
AH Transform (***Select only one.***)	ah-md5-hmac	AH with the MD5 (Message Digest 5) (a Hash-based Message Authentication Code [HMAC] variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (Secure Hash Algorithm) (an HMAC variant) authentication algorithm
ESP Encryption Transform (***Select only one.***)	esp-aes	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithim
	esp-aes 192	ESP with the 192-bit AES encryption algorithim.
	esp-aes 256	ESP with the 256-bit AES encryption algorithim
	esp-des	ESP with the 56-bit Data Encryption Standard (DES) encryption algorithm
	esp-3des	

	esp-null esp-seal	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES) Null encryption algorithm ESP with the 160-bit SEAL encryption algorithm.
ESP Authentication Transform (***Select only one.***)	esp-md5-hmac esp-sha-hmac	ESP with the MD5 (HMAC variant) authentication algorithm ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform	comp-lzs	IP compression with the Lempel-Ziv-Stac (LZS) algorithm

Step 2 (Optional) Change the mode associated with the transform set. The transport mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. The default mode is tunnel.

```
router(cfg-crypto-trans)# mode [transport | tunnel]
```

Syntax Description

tunnel | transport	(Optional) Specifies the mode for a transform set: either tunnel or transport mode If neither **tunnel** nor **transport** is specified, the default (tunnel mode) is assigned.

If you use this command to change the mode, the change will only affect the negotiation of subsequent IPsec SAs via crypto map entries that specify this transform set. (If you want the new settings to take effect sooner, you can clear all or part of the SA database.)

Editing Transform Sets

Complete the following steps if you need to edit a transform set:

Step 1 Delete the transform set from the crypto map.

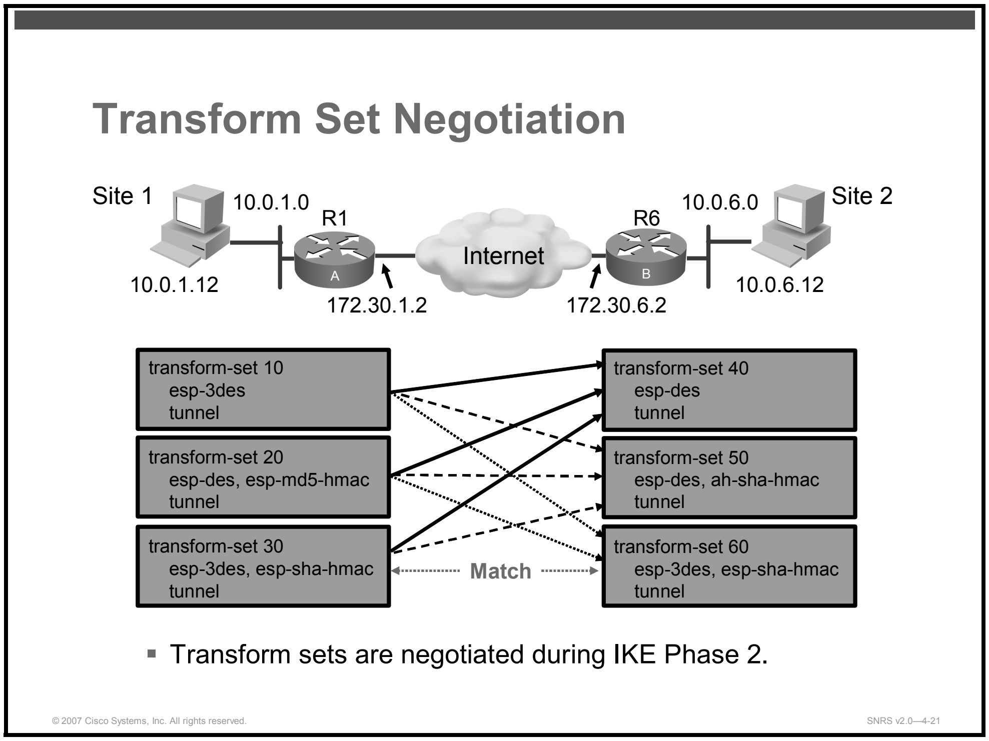
Step 2 Delete the transform set from the global configuration.

Step 3 Re-enter the transform set with corrections.

Step 4 Assign the transform set to a crypto map.

Step 5 Clear the SA database.

Step 6 Observe the SA negotiation and ensure that it works properly.



Transform Set Negotiation

transform-set 10
esp-3des
tunnel

transform-set 20
esp-des, esp-md5-hmac
tunnel

transform-set 30
esp-3des, esp-sha-hmac
tunnel

transform-set 40
esp-des
tunnel

transform-set 50
esp-des, ah-sha-hmac
tunnel

transform-set 60
esp-3des, esp-sha-hmac
tunnel

- Transform sets are negotiated during IKE Phase 2.

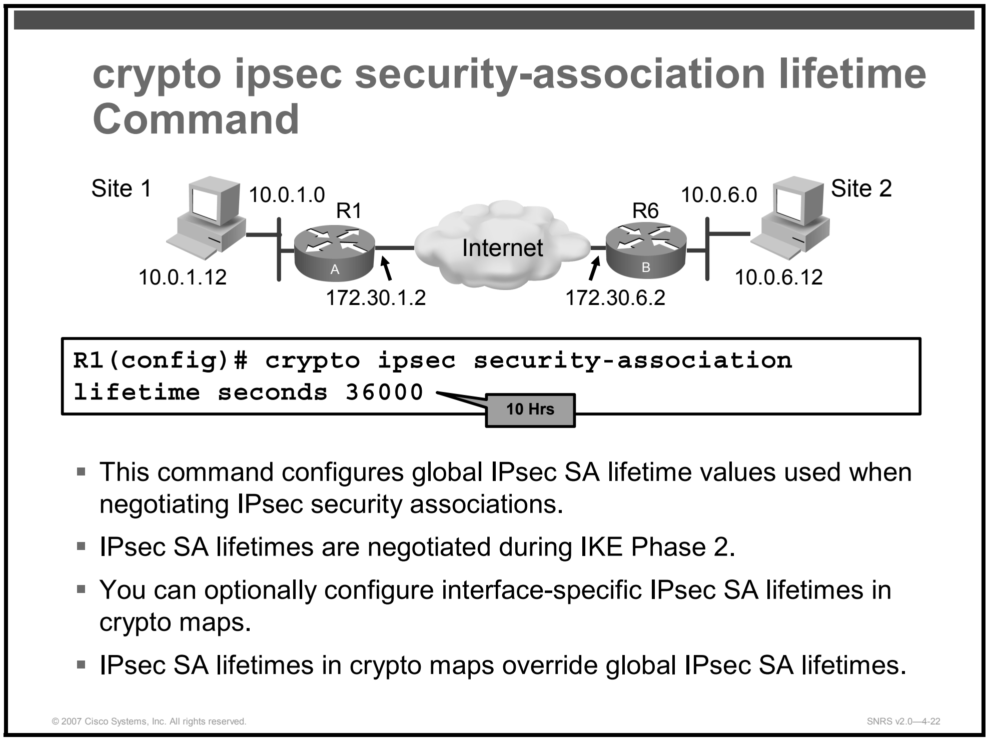
SNRS v2.0—4-21

Transform sets are negotiated during quick mode in IKE Phase 2 using the transform sets that you previously configured. You can configure multiple transform sets, and then specify one or more of the transform sets in a crypto map entry. Configure the transforms from most to least secure as per your policy. The transform set defined in the crypto map entry is used in the IPsec SA negotiation to protect the data flows specified by the ACL of that crypto map entry.

During the negotiation, the peers search for a transform set that is the same at both peers, as illustrated in the figure. Each of the R1 transform sets is compared against each of the R2 transform sets in succession. R1 transform sets 10, 20, and 30 are compared with R2 transform set 40. The result is no match. All of the R1 transform sets are then compared against the R2 transform sets. Ultimately, R1 transform set 30 matches R2 transform set 60. When such a transform set is found, it is selected and is applied to the protected traffic as part of the IPsec SAs of both peers. IPsec peers agree on one transform proposal per SA (unidirectional).

Configuring Global IPsec SA Lifetimes

This section describes how to configure global IPsec SA lifetimes.



crypto ipsec security-association lifetime Command

Site 1
10.0.1.0
R1
10.0.1.12
A
172.30.1.2
Internet
R6
10.0.6.0
Site 2
B
172.30.6.2
10.0.6.12

```
R1(config)# crypto ipsec security-association
lifetime seconds 36000
```

10 Hrs

- This command configures global IPsec SA lifetime values used when negotiating IPsec security associations.
- IPsec SA lifetimes are negotiated during IKE Phase 2.
- You can optionally configure interface-specific IPsec SA lifetimes in crypto maps.
- IPsec SA lifetimes in crypto maps override global IPsec SA lifetimes.

SNRS v2.0—4-22

You can change the global lifetime values that are used when negotiating new IPsec SAs.

Note These global lifetime values can be overridden for a particular crypto map entry.

These lifetimes only apply to SAs established via IKE. Manually established SAs do not expire.

There are two lifetimes: a timed lifetime and a traffic-volume lifetime. An SA expires after the first of these lifetimes is reached. The default lifetimes are 3600 seconds (1 hour) and 4,608,000 KB.

If you change a global lifetime, the new lifetime value will not be applied to the existing SAs, but will be used in the negotiation of subsequently established SAs. If you wish to use the new values immediately, you can clear all or part of the SA database. Use the **clear crypto sa** command to clear the SA database.

IPsec SAs use one or more shared-secret keys. These keys and their SAs time out together.

Follow these steps to configure global IPsec SA lifetimes:

- Change the global timed lifetime for IPsec SAs.

```
router(config)# crypto ipsec security-association lifetime {seconds
seconds | kilobytes kilobytes}
```

Syntax Description

seconds *seconds*	Specifies the number of seconds that an SA will live before expiring The default is 3600 seconds (1 hour).
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given SA before that SA expires The default is 4,608,000 KB.

IPsec SAs use shared-secret keys. These keys and their SAs time out together.

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new SAs during SA negotiation, it will specify its global lifetime value in the request to the peer; it will use this value as the lifetime of the new SAs. When the router receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new SAs.

There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. The security association expires after the first of these lifetimes is reached.

If you change a global lifetime, the change is only applied when the crypto map entry does not have a lifetime value specified. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command. Refer to the **clear crypto sa** command for more details.

To change the global timed lifetime, use the **crypto ipsec security-association lifetime seconds** form of the command. The timed lifetime causes the SA to time out after the specified number of seconds has passed.

To change the global traffic-volume lifetime, use the **crypto ipsec security-association lifetime kilobytes** form of the command. The traffic-volume lifetime causes the SA to time out after the specified amount of traffic (in kilobytes) has been protected by the key of the SA

Shorter lifetimes can make it harder to mount a successful key recovery attack, because the attacker has less data encrypted under the same key to work with. However, shorter lifetimes require more CPU processing time for establishing new SAs.

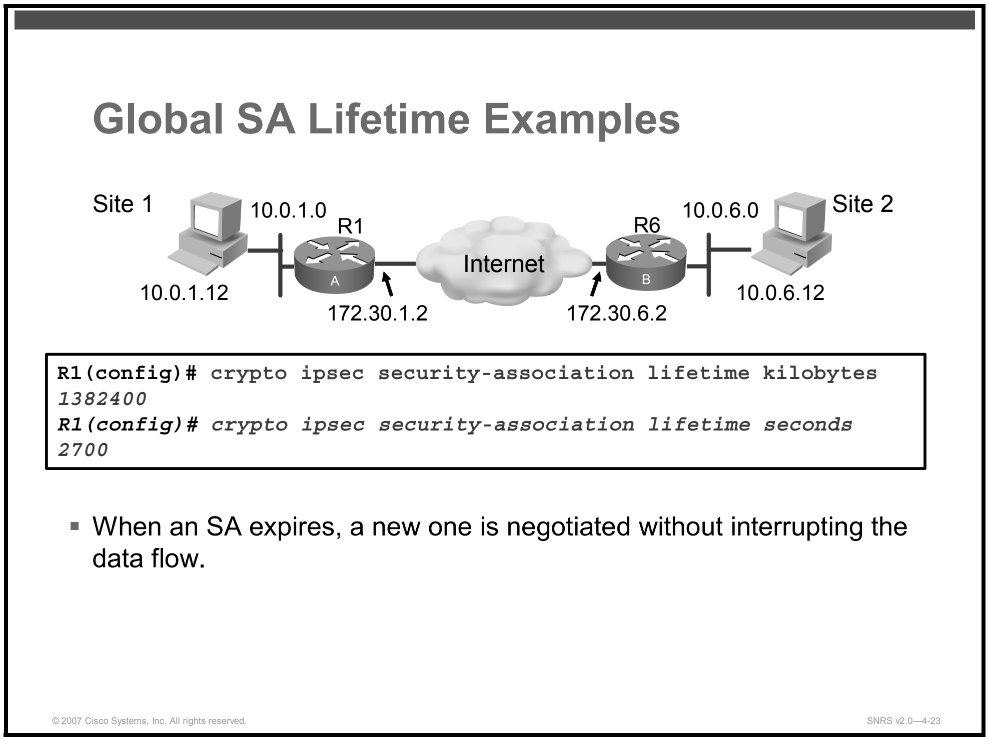
The lifetime values are ignored for manually established SAs (SAs installed using an **ipsec-manual** crypto map entry).

How These Lifetimes Work

The SA (and corresponding keys) will expire according to whichever occurs sooner, either after the number of seconds has passed (specified by the **seconds** keyword) or after the amount of traffic in kilobytes has passed (specified by the **kilobytes** keyword).

A new SA is negotiated *before* the lifetime threshold of the existing SA is reached, to ensure that a new SA is ready for use when the old one expires. The new SA is negotiated either 30 seconds before the seconds lifetime expires or when the volume of traffic through the tunnel reaches 256 KB less than the kilobytes lifetime (whichever occurs first).

If no traffic has passed through the tunnel during the entire life of the SA, a new SA is not negotiated when the lifetime expires. Instead, a new SA will be negotiated only when IPsec sees another packet that should be protected.



Global SA Lifetime Examples

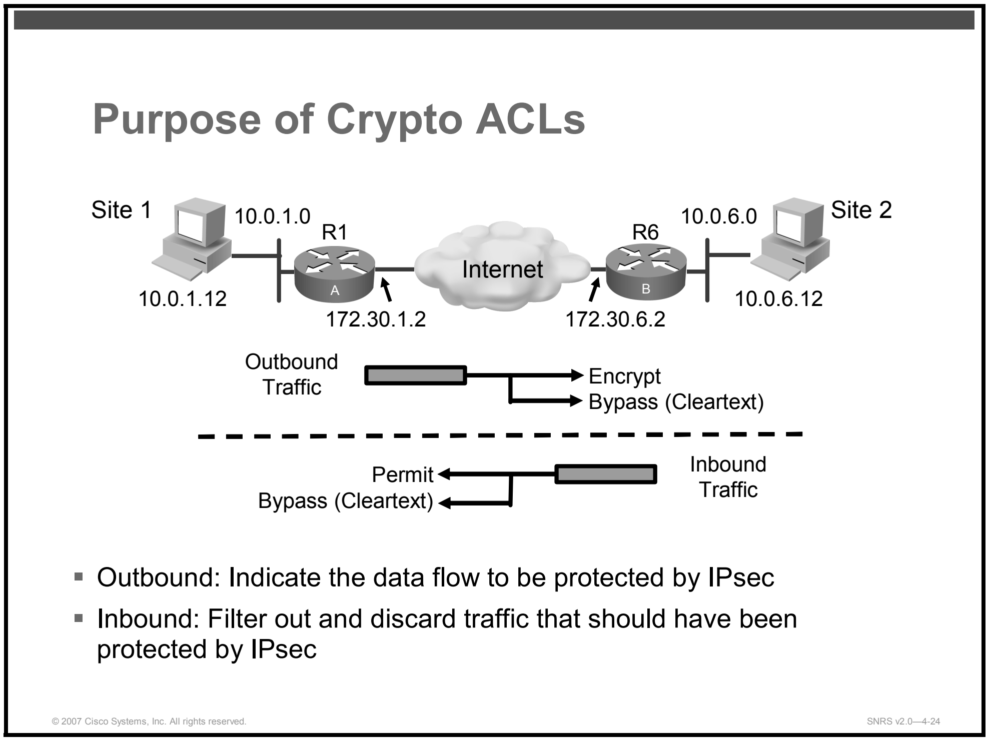
```
R1(config)# crypto ipsec security-association lifetime kilobytes
1382400
R1(config)# crypto ipsec security-association lifetime seconds
2700
```

- When an SA expires, a new one is negotiated without interrupting the data flow.

This figure shows an example of a global SA lifetime. The SA (and corresponding keys) will expire according to whichever occurs sooner, either after 2700 seconds (45 minutes) has passed or after 1,382,400 KB have passed.

Creating Crypto ACLs

This section describes how to create crypto ACLs.



Purpose of Crypto ACLs

- Outbound: Indicate the data flow to be protected by IPsec
- Inbound: Filter out and discard traffic that should have been protected by IPsec

Crypto ACLs define which IP traffic will be protected by encryption. Extended ACLs are used to specify further source and destination addresses and packet type.

Use encryption ACLs to control which packets on an interface are encrypted or decrypted, and which are transmitted as plain text (unencrypted).

When a packet is examined for an encryption ACL match, encryption ACL statements are checked in the order that the statements were created. After a packet matches the conditions in a statement, no more statements will be checked. This means that you need to carefully consider the order in which you enter the statements.

These ACL entries must mirror each other on the IPsec peers. If ACL entries include ranges of ports, a mirror image of those same ranges must be included on the remote peer ACL.

These ACLs are not the same as regular ACLs, which determine which traffic to forward or block at an interface. The ACLs themselves are not specific to IPsec. It is the crypto map entry referencing the specific ACL that defines whether IPsec processing is applied to the traffic matching a **permit** statement in the ACL.

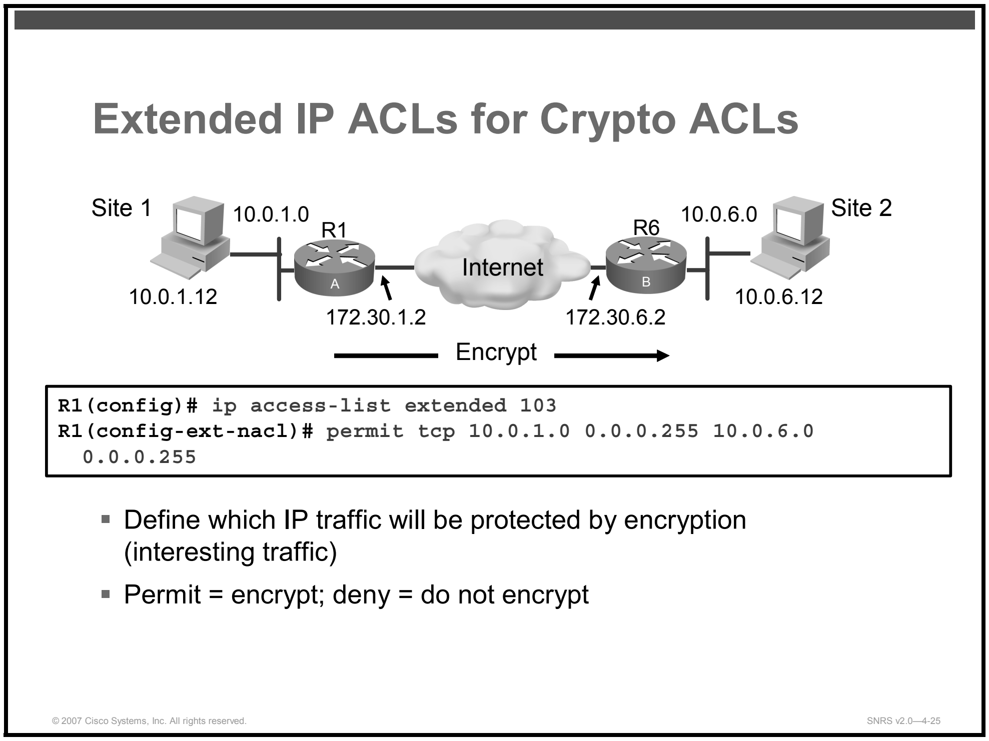
Crypto ACLs associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit equals protect)
- Indicate the data flow to be protected by the new SAs (specified by a single **permit** entry) when initiating negotiations for IPsec SAs
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec

- Determine whether to accept requests for IPsec SAs on behalf of the requested data flows when processing IKE negotiation from the IPsec peer (Negotiation is done only for **ipsec-isakmp** crypto map entries.) To be accepted, if the peer initiates the IPsec negotiation, it must specify a data flow that is permitted by a crypto ACL associated with an **ipsec-isakmp** crypto map entry.

If you want certain traffic to receive one combination of IPsec protection (for example, authentication only) and other traffic to receive a different combination of IPsec protection (for example, both authentication and encryption), you need to create two different crypto ACLs to define the two different types of traffic. These different ACLs are then used in different crypto map entries that specify different IPsec policies.

The crypto ACL that you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface.



Extended IP ACLs for Crypto ACLs

```
R1(config)# ip access-list extended 103
R1(config-ext-nacl)# permit tcp 10.0.1.0 0.0.0.255 10.0.6.0
  0.0.0.255
```

- Define which IP traffic will be protected by encryption (interesting traffic)
- Permit = encrypt; deny = do not encrypt

SNRS v2.0—4-25

To create an ACL, use the following commands:

- Define an encryption ACL by number and specify conditions to determine which IP packets will be protected. (Or follow the next step.)

```
router(config)# access-list access-list-number [dynamic dynamic-
name [timeout minutes]] {deny | permit} protocol source source-
wildcard destination destination-wildcard [log]
```

Syntax Description

access-list-number	Number of an encryption ACL. This is a decimal number from 100 to 199.
dynamic *dynamic-name*	(Optional) Identifies this encryption ACL as a dynamic encryption ACL. Refer to lock-and-key access documented in the "Configuring Lock-and-Key Security (Dynamic Access Lists)" chapter in the *Cisco IOS Security Configuration Guide.*
timeout *minutes*	(Optional) Specifies the absolute length of time (in minutes) that a temporary ACL entry can remain in a dynamic ACL. The default is an infinite length of time and allows an entry to remain permanently. Refer to lock-and-key access documented in the "Configuring Lock-and-Key Security (Dynamic Access Lists)" chapter in the *Cisco IOS Security Configuration Guide.*
deny	Does not encrypt or decrypt IP traffic if the conditions are matched.
permit	Encrypts and decrypts IP traffic if the conditions are matched.
protocol	Name or number of an IP protocol. It can be one of the keywords **eigrp**, **gre**, **icmp**, **igmp**, **igrp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or an integer in the range of 0 to 255 representing an IP protocol number. To match any Internet protocol, including ICMP, TCP, and UDP, use the keyword **ip**. Some protocols allow further qualifiers, as described in text that follows.
source	Number of the network or host from which the packet is being sent. There are three other ways to specify the source: ■ Use a 32-bit quantity in four-part dotted decimal format. ■ Use the keyword **any** as an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section "Usage Guidelines"). ■ Use the **host** source as an abbreviation for a source and source wildcard of source 0.0.0.0.
source-wildcard	Wildcard bits (mask) to be applied to source. There are three other ways to specify the source wildcard: ■ Use a 32-bit quantity in four-part, dotted decimal format. Place 1s in the bit positions that you want to ignore. ■ Use the keyword **any** as an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section "Usage Guidelines"). ■ Use the **host** source as an abbreviation for a source and source wildcard of source 0.0.0.0.
destination	Number of the network or host to which the packet is being sent. There are three other ways to specify the destination: ■ Use a 32-bit quantity in four-part, dotted decimal format. ■ Use the keyword **any** as an abbreviation for the destination and destination wildcard of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section "Usage Guidelines"). ■ Use the **host** destination as an abbreviation for a destination and destination wildcard of destination 0.0.0.0.

destination-wildcard	Wildcard bits to be applied to the destination. There are three other ways to specify the destination wildcard: ■ Use a 32-bit quantity in four-part, dotted decimal format. Place 1s in the bit positions that you want to ignore. ■ Use the keyword **any** as an abbreviation for a destination and destination wildcard of 0.0.0.0 255.255.255.255. This keyword is normally not recommended (see the section "Usage Guidelines"). ■ Use the **host** destination as an abbreviation for a destination and destination wildcard of destination 0.0.0.0.
icmp-type	(Optional) ICMP packets can be matched for encryption by ICMP message type. The type is a number from 0 to 255.
icmp-code	(Optional) ICMP packets that are matched for encryption by ICMP message type can also be matched by the ICMP message code. The code is a number from 0 to 255.
icmp-message	(Optional) ICMP packets can be matched for encryption by an ICMP message type name or ICMP message type and code name. The possible names are discussed in the section "Usage Guidelines."
igmp-type	(Optional) IGMP packets can be matched for encryption by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the section "Usage Guidelines."
operator	(Optional) Compares source or destination ports. Possible operators include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the source and source wildcard, it must match the source port. If the operator is positioned after the destination and destination wildcard, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number.
port	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP port names are listed in the section "Usage Guidelines." TCP port names can be used only when filtering TCP. UDP port names are listed in the section "Usage Guidelines." UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: indicates an established connection. A match occurs if the TCP datagram has the acknowledgment (ACK) or reset (RST) bits set. The nonmatching case is that of the initial TCP datagram to form a connection.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) The message includes the ACL number, regardless of whether the packet was encrypted or decrypted; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets encrypted or decrypted in the prior 5-minute interval.

- Create an extended ACL and specify conditions to determine which IP packets will be protected.

```
router(config)# ip access-list extended access-list-number
[dynamic dynamic-name [timeout minutes]] {deny | permit} protocol
source source-wildcard destination destination-wildcard [log]
```

— Then follow with **permit** and **deny** statements as appropriate.

Note Although the ACL syntax is unchanged, the meanings are slightly different for crypto ACLs. The **permit** statement specifies that matching packets must be encrypted; **deny** specifies that matching packets need not be encrypted.

Using the **permit** keyword causes all IP traffic that matches the specified conditions to be protected using the policy described by the corresponding crypto map entry. Using the **deny** keyword prevents traffic from being protected in the context of that particular crypto map

If this traffic is denied in all of the crypto map entries for that interface, the traffic is not protected.

If you configure multiple statements for a given crypto ACL that is used for IPsec, in general the first **permit** statement that is matched will be the statement used to determine the scope of the IPsec SA. That is, the IPsec SA will be set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto ACL, a new, separate IPsec SA will be negotiated to protect traffic matching the newly matched ACL statement.

Note ACLs for crypto map entries tagged as **ipsec-manual** are restricted to a single **permit** entry and subsequent entries are ignored. In other words, the SAs established by that particular crypto map entry are only for a single data flow. To be able to support multiple manually established SAs for different kinds of traffic, define multiple crypto ACLs and then apply each one to a separate **ipsec-manual** crypto map entry. Each ACL should include one **permit** statement defining what traffic to protect.

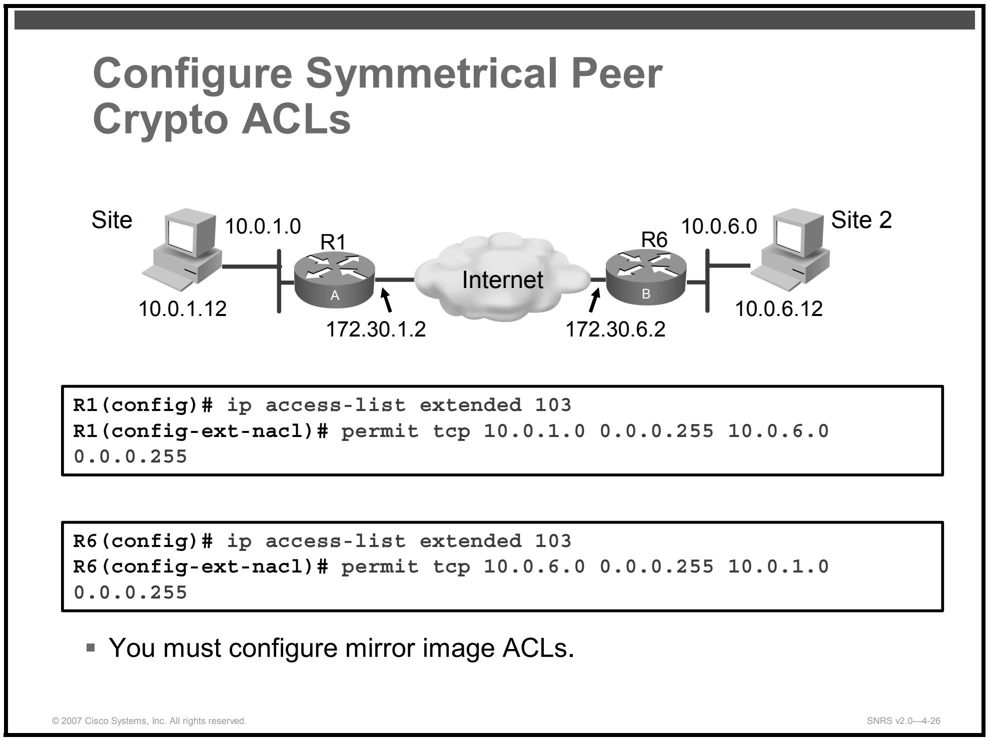
Any unprotected inbound traffic that matches a **permit** entry in the crypto ACL for a crypto map entry flagged as IPsec will be dropped, because this traffic was expected to be protected by IPsec.

In a later step, you will associate a crypto ACL to a crypto map, which, in turn, is assigned to a specific interface.

Defining Mirror Image Crypto ACLs at Each IPsec Peer

It is recommended that for every crypto ACL specified for a static crypto map entry that you define at the local peer, you define a mirror image crypto ACL at the remote peer. This practice ensures that traffic that has IPsec protection applied locally can be processed correctly at the remote peer.

Note The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.



Configure Symmetrical Peer Crypto ACLs

```
R1(config)# ip access-list extended 103
R1(config-ext-nacl)# permit tcp 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
```

```
R6(config)# ip access-list extended 103
R6(config-ext-nacl)# permit tcp 10.0.6.0 0.0.0.255 10.0.1.0
0.0.0.255
```

- You must configure mirror image ACLs.

SNRS v2.0—4-26

You must configure mirror image crypto ACLs for use by IPsec. Both inbound and outbound traffic is evaluated against the same outbound IPsec ACL. The criteria of the ACL are applied in the forward direction to traffic exiting your router and the reverse direction to traffic entering your router. When a router receives encrypted packets back from an IPsec peer, it uses the same ACL to determine which inbound packets to decrypt by viewing the source and destination addresses in the ACL in reverse order.

The example shown in the figure illustrates why symmetrical ACLs are recommended. For site 1, IPsec protection is applied to traffic between hosts on the 10.0.1.0 network as the data exits the router A serial interface and route to site 2 hosts on the 10.0.6.0 network. For traffic from site 1 hosts on the 10.0.1.0 network to site 2 hosts on the 10.0.6.0 network, the ACL entry on router A is evaluated as follows:

- Source = Hosts on 10.0.1.0 network
- Destination = Hosts on 10.0.6.0 network

For incoming traffic from site 2 hosts on the 10.0.6.0 network to site 1 hosts on the 10.0.1.0 network, that same ACL entry on router A is evaluated as follows:

- Source = Hosts on 10.0.6.0 network
- Permit = Hosts on 10.0.1.0 network

Using the any Keyword in Crypto ACLs

Using the **any** keyword could cause problems when you create crypto ACLs.

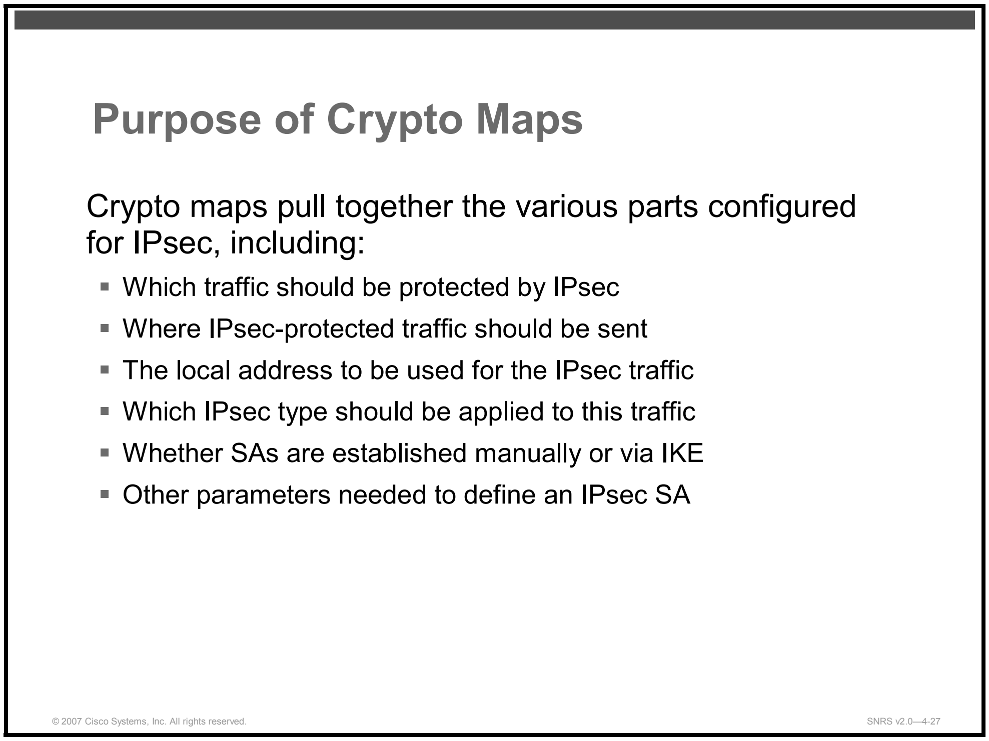
Caution It is recommended that you avoid using the **any** keyword to specify source or destination addresses.

The **permit any any** statement is also strongly discouraged, because that statement will cause all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and will require protection for all inbound traffic. Then, all inbound packets that lack IPsec protection will be silently dropped, including packets for routing protocols, Network Time Protocol (NTP), echo, echo response, and so on.

You need to be sure that you define which packets to protect. If you must use the **any** keyword in a **permit** statement, you must preface that statement with a series of **deny** statements to filter out any traffic (that would otherwise fall within that permit statement) that you do not want to be protected.

Creating Crypto Maps

This section describes how to create and configure IPsec crypto maps.



Purpose of Crypto Maps

Crypto maps pull together the various parts configured for IPsec, including:

- Which traffic should be protected by IPsec
- Where IPsec-protected traffic should be sent
- The local address to be used for the IPsec traffic
- Which IPsec type should be applied to this traffic
- Whether SAs are established manually or via IKE
- Other parameters needed to define an IPsec SA

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-27

Crypto map entries must be created for IPsec to set up SAs for traffic flows that must be encrypted.

Crypto map entries created for IPsec set up SA parameters, tying together the various parts configured for IPsec, including these:

- Which traffic should be protected by IPsec (per a crypto ACL)
- Where IPsec-protected traffic should be sent (who the remote IPsec peer is)
- The local address to be used for the IPsec traffic
- Which IPsec security type should be applied to this traffic (transform sets)
- Whether SAs are established manually or are established via IKE
- Other parameters that might be necessary to define an IPsec SA

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, an SA is negotiated with the remote peer according to the parameters included in the crypto map entry. Otherwise, if the crypto map entry specifies the use of manual SAs, an SA should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no SA exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of SAs. If the local router initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified IPsec peer. If the IPsec peer initiates the negotiation, the local router will check the policy from the static crypto map entries and any referenced dynamic crypto map entries to decide whether to accept or reject the request (offer) from the peer.

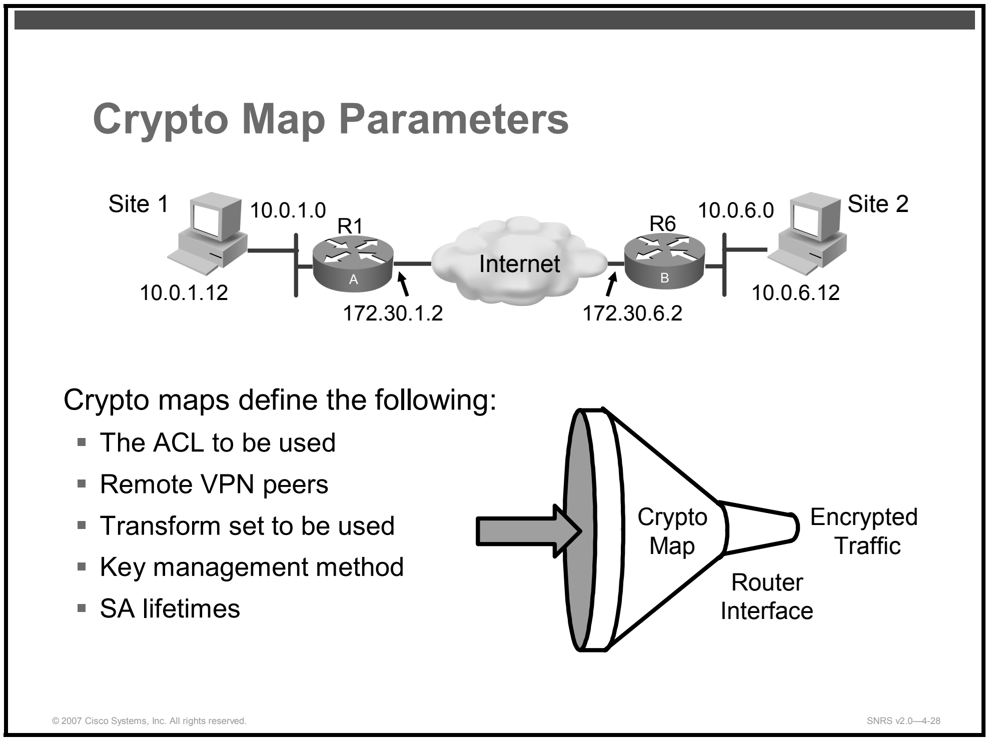
For IPsec to succeed between two IPsec peers, the crypto map entries of both peers must contain compatible configuration statements.

When two peers try to establish an SA, they must each have at least one crypto map entry that is compatible with one of the crypto map entries of the other peer. For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto ACLs (for example, mirror image ACLs). In the case where the responding peer is using dynamic crypto maps, the entries in the local crypto ACL must be permitted by the peer crypto ACL.
- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

Load Sharing

You can define multiple remote peers using crypto maps to allow for load sharing. If one peer fails, there will still be a protected path. The peer to which packets are actually sent is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.



Crypto Map Parameters

Crypto maps define the following:

- The ACL to be used
- Remote VPN peers
- Transform set to be used
- Key management method
- SA lifetimes

SNRS v2.0—4-28

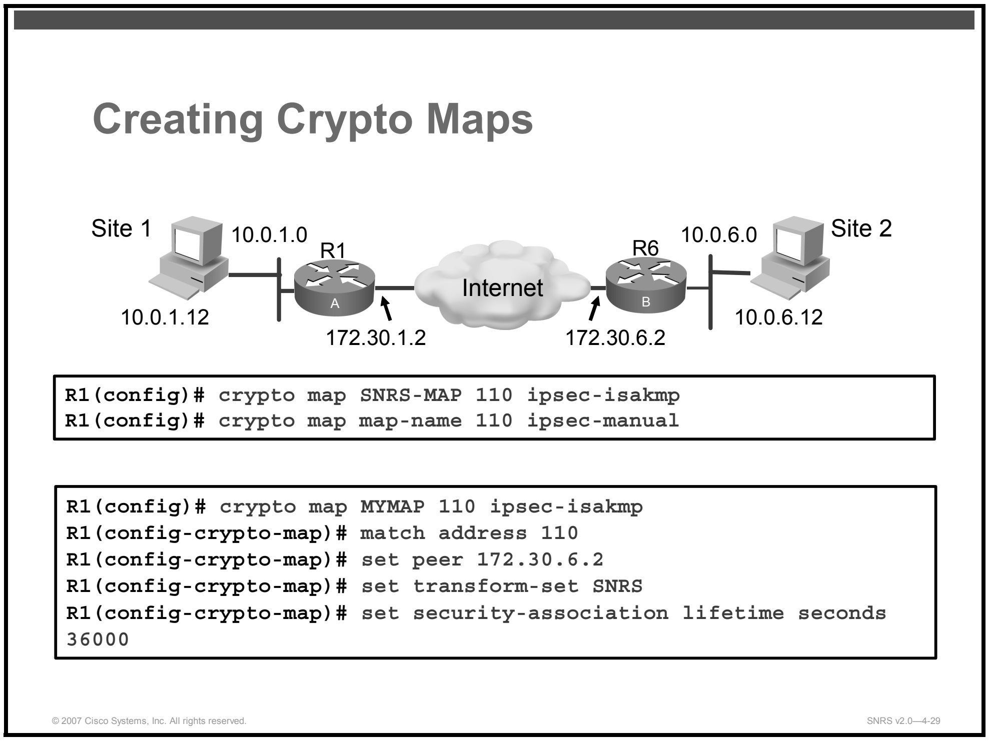
You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPsec using IKE, and IPsec with manually configured SA entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

How Many Crypto Maps Should You Create?

If you create more than one crypto map entry for a given interface, use the sequence number (*seq-num*) of each map entry to rank the map entries: the lower the sequence number, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher-priority map entries first.

You must create multiple crypto map entries for a given interface if any of the following conditions exist:

- If different data flows are to be handled by separate IPsec peers.
- If you want to apply different IPsec security to different types of traffic (to the same or separate IPsec peers); for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case, the different types of traffic should have been defined in two separate ACLs, and you must create a separate crypto map entry for each crypto ACL.
- If you are not using IKE to establish a particular set of SAs, and want to specify multiple ACL entries, you must create separate ACLs (one per permit entry) and specify a separate crypto map entry for each ACL.



Creating Crypto Maps

```
R1(config)# crypto map SNRS-MAP 110 ipsec-isakmp
R1(config)# crypto map map-name 110 ipsec-manual
```

```
R1(config)# crypto map MYMAP 110 ipsec-isakmp
R1(config-crypto-map)# match address 110
R1(config-crypto-map)# set peer 172.30.6.2
R1(config-crypto-map)# set transform-set SNRS
R1(config-crypto-map)# set security-association lifetime seconds
36000
```

SNRS v2.0—4-29

Creating Crypto Maps

When IKE is used to establish SAs, the IPsec peers can negotiate the settings that they will use for the new SAs. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

Follow these steps to create crypto map entries that will use IKE to establish the SAs:

Step 1 Name the crypto map to create, specify ISAKMP SAs, and enter crypto map configuration mode.

```
router(config)#crypto map map-name seq-num ipsec-manual
```

Or

```
router(config)#crypto map map-name seq-num ipsec-isakmp
[dynamic dynamic-map-name] [discover]
```

Syntax Description

map-name	The name you assign to the crypto map set
seq-num	The number you assign to the crypto map entry
ipsec-manual	Indicates that IKE will not be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry
ipsec-isakmp	Indicates that IKE will be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry
dynamic	(Optional) Specifies that this crypto map entry is to reference a pre-existing dynamic crypto map Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPsec device. If you use this keyword, none of the crypto map configuration commands will be available.

dynamic-map-name	(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template
discover	(Optional) Enables peer discovery By default, peer discovery is not enabled.

Step 2 Name an IPsec crypto ACL.

```
router(config-crypto-map)# match address [access-list-id |
name]
```

Syntax Description

access-list-id	(Optional) Identifies the extended ACL by its name or number This value should match the *access-list-number* or *name* argument of the extended ACL being matched.
name	(Optional) Identifies the named encryption ACL This name should match the *name* argument of the named encryption ACL being matched.

This command is required for all static crypto map entries. If you are defining a dynamic crypto map entry (with the **crypto dynamic-map** command), this command is not required but is strongly recommended.

Use this command to assign an extended ACL to a crypto map entry. You also need to define this ACL using the **access-list** or **ip access-list extended** commands.

The extended ACL specified with this command will be used by IPsec to determine which traffic should be protected and which traffic does not need protection. (Traffic that is permitted by the ACL will be protected. Traffic that is denied by the ACL will not be protected in the context of the corresponding crypto map entry.)

Note that the crypto ACL is not used to determine whether to permit or deny traffic through the interface. An ACL applied directly to the interface makes that determination.

The crypto ACL specified by this command is used when evaluating both inbound and outbound traffic. Outbound traffic is evaluated against the crypto ACLs specified by the interface crypto map entries to determine if it should be protected and if so (if traffic matches a **permit** entry) which crypto policy applies. (If necessary, in the case of static IPsec crypto maps, new SAs are established using the data flow identity as specified in the **permit** entry; in the case of dynamic crypto map entries, if no SA exists, the packet is dropped.) After passing the regular ACLs at the interface, inbound traffic is evaluated against the crypto ACLs specified by the entries of the interface crypto map set to determine if it should be protected and, if so, which crypto policy applies. (In the case of IPsec, unprotected traffic is discarded because it should have been protected by IPsec.)

In the case of IPsec, the ACL is also used to identify the flow for which the IPsec SAs are established. In the outbound case, the **permit** entry is used as the data flow identity (in general), while in the inbound case, the data flow identity specified by the peer must be permitted by the crypto ACL.

Step 3 Specify the remote IPsec peer.

```
router(config-crypto-map)# set peer {host-name [dynamic]
[default] | ip-address [default]}
```

Syntax Description

host-name	This specifies the IPsec peer by its host name. This is the hostname of the peer concatenated with its domain name (for example, myhost.example.com).
dynamic	(Optional) The host name of the IPsec peer will be resolved via a DNS lookup right before the router establishes the IPsec tunnel.
default	(Optional) If there are multiple IPsec peers, this keyword designates that the first peer is the default peer.
ip-address	This specifies the IPsec peer by its IP address.

Use this command to specify an IPsec peer for a crypto map.

This command is required for all static crypto maps. If you are defining a dynamic crypto map (with the **crypto dynamic-map** command), this command is not required, and in most cases is not used (because, in general, the peer is unknown).

For crypto map entries created with the **crypto map** *map-name seq-num* **ipsec-isakmp** command, you can specify multiple peers by repeating this command. The peer to which packets are actually sent is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

For crypto map entries created with the **crypto map** *map-name seq-num* **ipsec-manual** command, you can specify only one IPsec peer per crypto map. If you want to change the peer, you must first delete the old peer and then specify the new peer.

You can specify the remote IPsec peer by its hostname only if the hostname is mapped to the IP address of the peer in a DNS or if you manually map the hostname to the IP address with the **ip host** command.

The dynamic Keyword

When specifying the hostname of a remote IPsec peer via the **set peer** command, you can also issue the **dynamic** keyword, which defers DNS resolution of the hostname until right before the IPsec tunnel has been established. Deferring resolution enables the Cisco IOS Software to detect whether the IP address of the remote IPsec peer has changed. Thus, the Cisco IOS Software can contact the peer at the new IP address.

If the **dynamic** keyword is not issued, the hostname is resolved immediately after it is specified. So, the Cisco IOS Software cannot detect an IP address change and, therefore, attempts to connect to the IP address that it previously resolved.

The default Keyword

If there are multiple peers and you specify the **default** keyword, the first peer is designated as the default peer.

If dead peer detection (DPD) detects a failure, the default peer is retried before there is an attempt to connect to the next peer in the peer list.

If the default peer is unresponsive, the next peer in the peer list becomes the new current peer. Future connections through the crypto map will try that peer.

Step 4 Specify which transform set should be used.

```
router(config-crypto-map)# set transform-set transform-set-
name [transform-set-name2...transform-set-name6]
```

Syntax Description

transform-set-name	Name of the transform set For an **ipsec-manual** crypto map entry, you can specify only one transform set. For an **ipsec-isakmp** or dynamic crypto map entry, you can specify up to six transform sets.

This command is required for all static and dynamic crypto map entries.

Use this command to specify which transform sets to include in a crypto map entry.

For an **ipsec-isakmp** crypto map entry, you can list multiple transform sets with this command. List the higher priority transform sets first.

If the local router initiates the negotiation, the transform sets are presented to the peer in the order specified in the crypto map entry. If the peer initiates the negotiation, the local router accepts the first transform set that matches one of the transform sets specified in the crypto map entry.

The first matching transform set that is found at both peers is used for the SA. If no match is found, IPsec will not establish an SA. The traffic will be dropped because there is no SA to protect the traffic.

For an **ipsec-manual** crypto map entry, you can specify only one transform set. If the transform set does not match the transform set at the crypto map of the remote peer, the two peers will fail to correctly communicate because the peers are using different rules to process the traffic.

If you want to change the list of transform sets, re-specify the new list of transform sets to replace the old list. This change is only applied to crypto map entries that reference this transform set. The change will not be applied to existing SAs, but will be used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command.

Any transform sets included in a crypto map must previously have been defined using the **crypto ipsec transform-set** command.

You can list multiple transforms in order of priority (highest priority first).

Step 5 (Optional) Override (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPsec SAs.

```
router(config-crypto-map)# set security-association lifetime
{seconds <seconds | kilobytes kilobytes}>
```

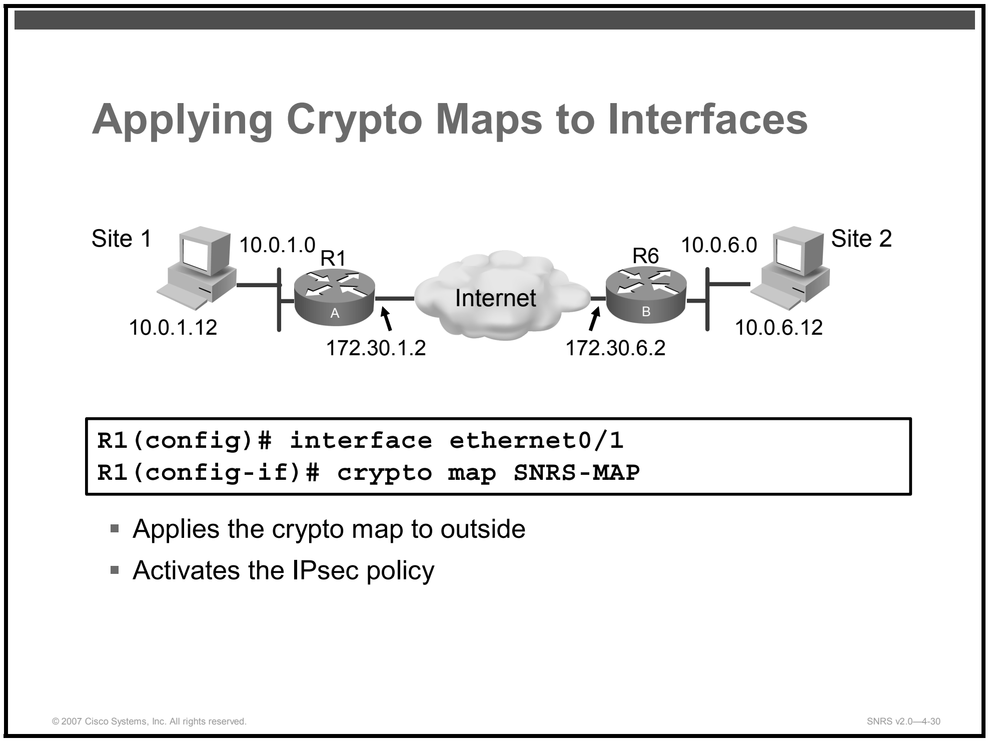
Syntax Description

seconds *seconds*	Specifies the number of seconds that an SA will live before expiring
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given SA before that SA expires

This command is available only for **ipsec-isakmp** crypto map entries and dynamic crypto map entries.

Applying Crypto Maps to Interfaces

This topic describes how to apply crypto maps to interfaces.



Applying Crypto Maps to Interfaces

```
R1(config)# interface ethernet0/1
R1(config-if)# crypto map SNRS-MAP
```

- Applies the crypto map to outside
- Activates the IPsec policy

A crypto map set will need to be applied to each interface through which IPsec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all of the interface traffic against the crypto map set and to use the specified policy during connection or SA negotiation on behalf of traffic to be encrypted.

Use the following command to assign a crypto map set to an interface:

router(config-if)# **crypto map** *map-name* [**redundancy** *standby-group-name*[**stateful**]]

Syntax Description

map-name	Name that identifies the crypto map set (This is the name assigned when the crypto map was created.) When the **no** form of the command is used, this argument is optional. Any value supplied for the argument is ignored.
redundancy	(Optional) Defines a backup IPsec peer Both routers in the standby group are defined by the redundancy *standby name* and share the same virtual IP address.
standby-group-name	(Optional) Refers to the name of the standby group as defined by Hot Standby Router Protocol (HSRP) standby commands
stateful	(Optional) Enables IPsec stateful failover for the crypto map

You must assign a crypto map set to an interface before that interface can provide IPsec services. Only one crypto map set can be assigned to an interface. If multiple crypto map entries have the same map name but a different sequence number, they are considered to be part of the same set and will all be applied to the interface. The crypto map entry that has the lowest sequence number is considered the highest priority and will be evaluated first. A single crypto map set can contain a combination of **ipsec-isakmp** and **ipsec-manual crypto map** entries.

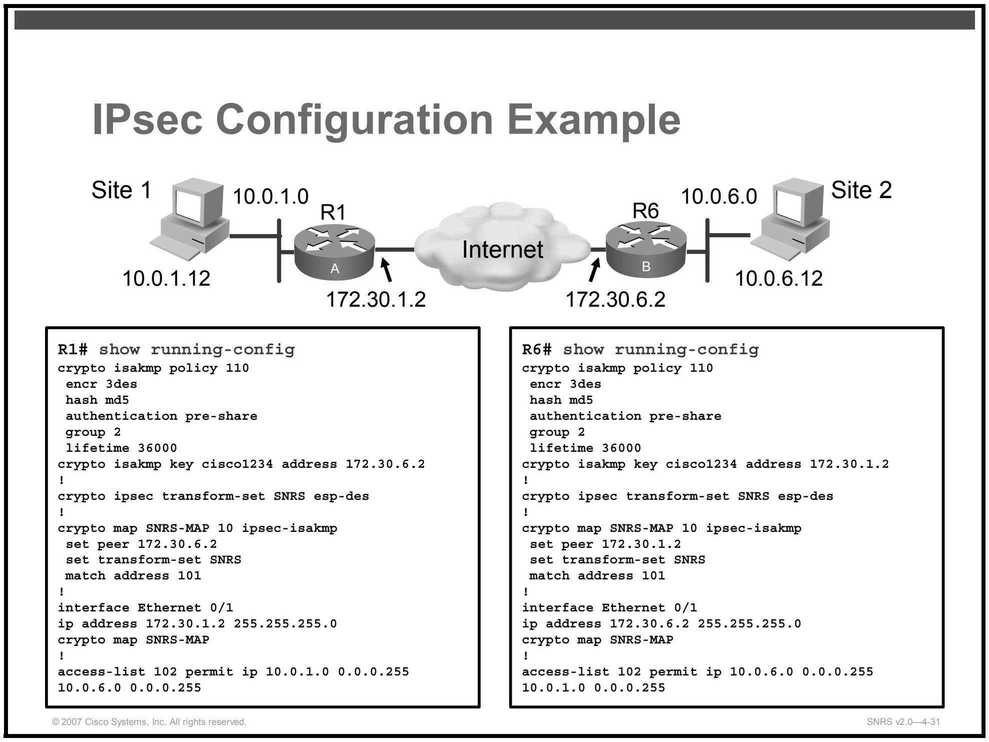
The standby name must be configured on all devices in the standby group, and the standby address must be configured on at least one member of the group. If the standby name is removed from the router, the IPsec SAs will be deleted. If the standby name is added again, regardless of whether the same name or a different name is used, the crypto map (using the **redundancy** option) will have to be reapplied to the interface.

Note A virtual IP address must be configured in the standby group to enable either stateless or stateful redundancy.

The **stateful** keyword enables stateful failover of IKE and IPsec sessions.

IPsec Configuration Example

This section gives an example of an IPsec configuration for a site-to-site VPN using pre-shared keys.



IPsec Configuration Example

```
R1# show running-config
crypto isakmp policy 110
 encr 3des
 hash md5
 authentication pre-share
 group 2
 lifetime 36000
crypto isakmp key cisco1234 address 172.30.6.2
!
crypto ipsec transform-set SNRS esp-des
!
crypto map SNRS-MAP 10 ipsec-isakmp
 set peer 172.30.6.2
 set transform-set SNRS
 match address 101
!
interface Ethernet 0/1
ip address 172.30.1.2 255.255.255.0
crypto map SNRS-MAP
!
access-list 102 permit ip 10.0.1.0 0.0.0.255
10.0.6.0 0.0.0.255
```

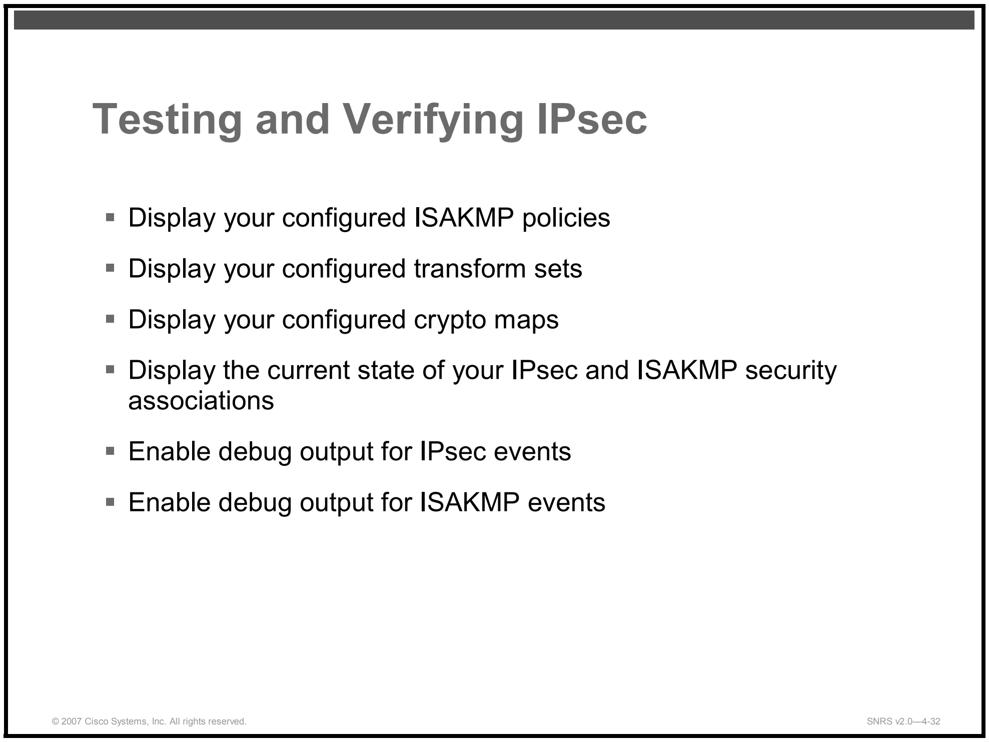
```
R6# show running-config
crypto isakmp policy 110
 encr 3des
 hash md5
 authentication pre-share
 group 2
 lifetime 36000
crypto isakmp key cisco1234 address 172.30.1.2
!
crypto ipsec transform-set SNRS esp-des
!
crypto map SNRS-MAP 10 ipsec-isakmp
 set peer 172.30.1.2
 set transform-set SNRS
 match address 101
!
interface Ethernet 0/1
ip address 172.30.6.2 255.255.255.0
crypto map SNRS-MAP
!
access-list 102 permit ip 10.0.6.0 0.0.0.255
10.0.1.0 0.0.0.255
```

SNRS v2.0—4-31

Consider these configuration examples for R1 and R6 in the figure. The examples are concatenated to show only commands related to what has been covered in this lesson to this point.

Testing and Verifying IPsec

This topic describes the commands used to test and verify IPsec configurations.



Testing and Verifying IPsec

- Display your configured ISAKMP policies
- Display your configured transform sets
- Display your configured crypto maps
- Display the current state of your IPsec and ISAKMP security associations
- Enable debug output for IPsec events
- Enable debug output for ISAKMP events

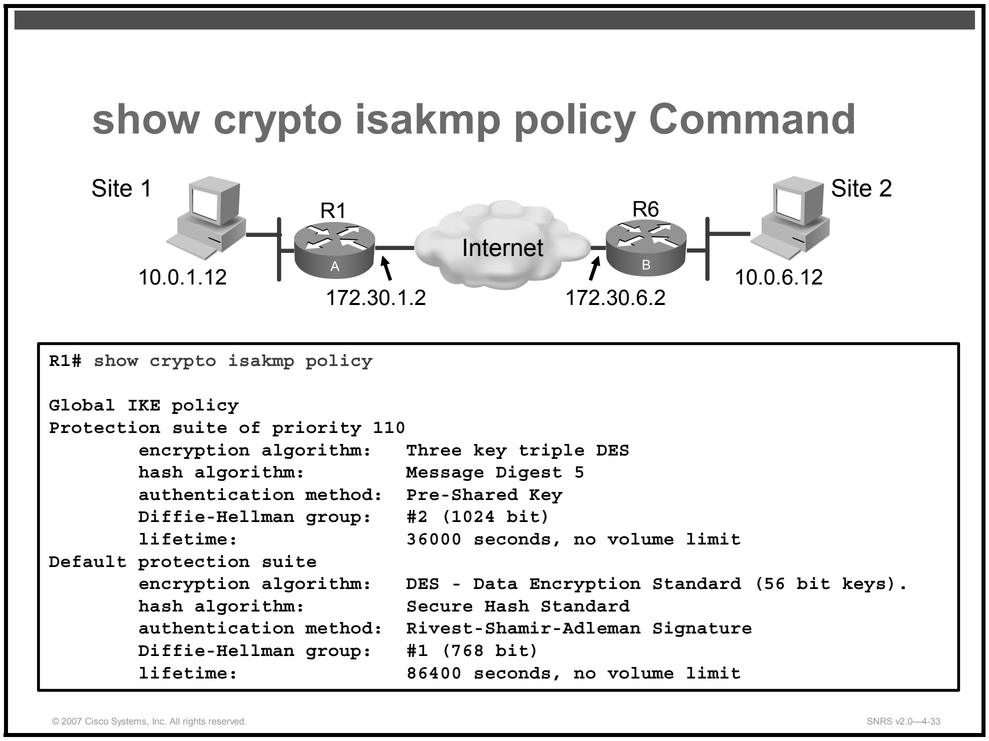
SNRS v2.0—4-32

You can perform the following actions to test and verify that you have correctly configured the IPsec site-to-site VPN:

- Display your configured ISAKMP policies using the **show crypto isakmp policy** command
- Display your configured transform sets using the **show crypto ipsec transform-set** command
- Display the current state of your ISAKMP SAs with the **show crypto isakmp sa** command
- Display the current state of your IPsec SAs with the **show crypto ipsec sa** command
- View your configured crypto maps with the **show crypto map** command
- Debug ISAKMP and IPsec traffic through the Cisco IOS Software with the **debug crypto ipsec** and **debug crypto isakmp** commands

Displaying ISAKMP Policies

This section describes how to display ISAKMP policies.



show crypto isakmp policy Command

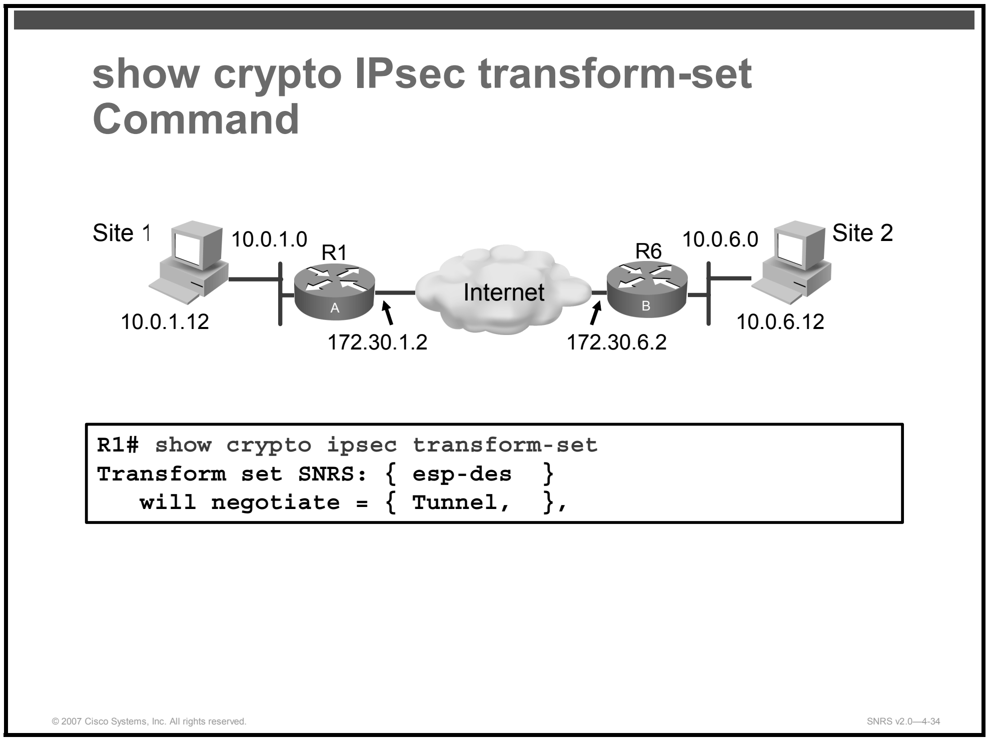
```
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 110
        encryption algorithm:   Three key triple DES
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               36000 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
```

Use the **show crypto isakmp policy** command to view the parameters for each ISAKMP policy, as shown in the example.

```
router# show crypto isakmp policy
```

Displaying Transform Sets

This section describes how to display configured transform sets.



show crypto IPsec transform-set Command

```
R1# show crypto ipsec transform-set
Transform set SNRS: { esp-des  }
   will negotiate = { Tunnel,  },
```

SNRS v2.0—4-34

Use the **show crypto ipsec transform-set** command to view the configured transform sets.

```
router# show crypto ipsec transform-set [transform-set-name]
```

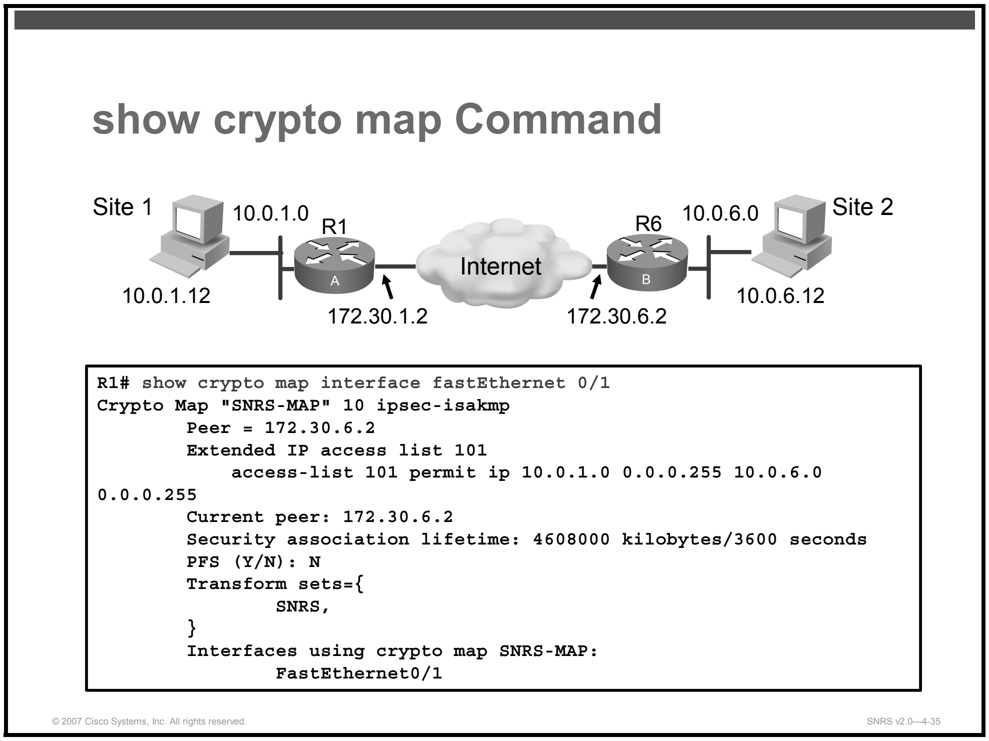
Syntax Description

transform-set-name	(Optional) Only the transform sets with the specified *transform-set-name* are displayed.

If no keyword is used, all transform sets configured at the router are displayed.

Displaying Crypto Maps

This section describes how to view configured crypto maps.



show crypto map Command

```
R1# show crypto map interface fastEthernet 0/1
Crypto Map "SNRS-MAP" 10 ipsec-isakmp
        Peer = 172.30.6.2
        Extended IP access list 101
            access-list 101 permit ip 10.0.1.0 0.0.0.255 10.0.6.0
0.0.0.255
        Current peer: 172.30.6.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                SNRS,
        }
        Interfaces using crypto map SNRS-MAP:
                FastEthernet0/1
```

The **show crypto map** command is used to view the crypto map configuration. If no keywords are used, all crypto maps configured at the router will be displayed.

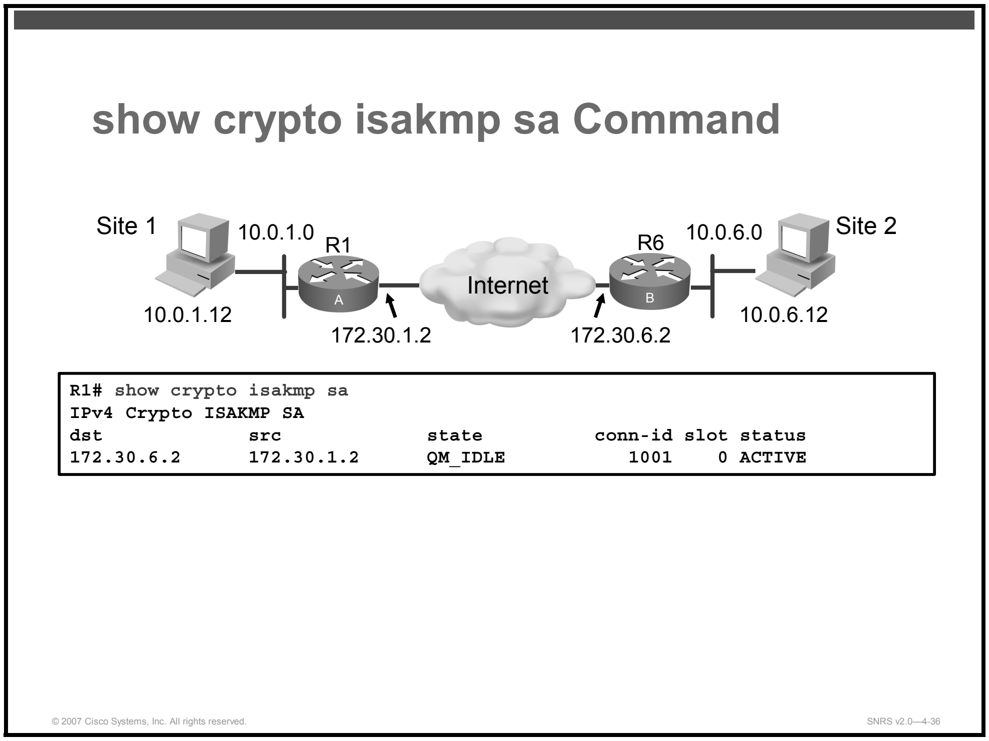
router# **show crypto map** [**interface** *interface* | **tag** *map-name*]

Syntax Description

interface *interface*	(Optional) Displays only the crypto map set that is applied to the specified interface
tag *map-name*	(Optional) Displays only the crypto map set with the specified *map-name*

Displaying Current State of ISAKMP and IPsec SAs

This section describes how to display the current state of ISAKMP and IPsec SAs.



show crypto isakmp sa Command

```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.6.2      172.30.1.2      QM_IDLE           1001    0 ACTIVE
```

© 2007 Cisco Systems, Inc. All rights reserved.   SNRS v2.0—4-36

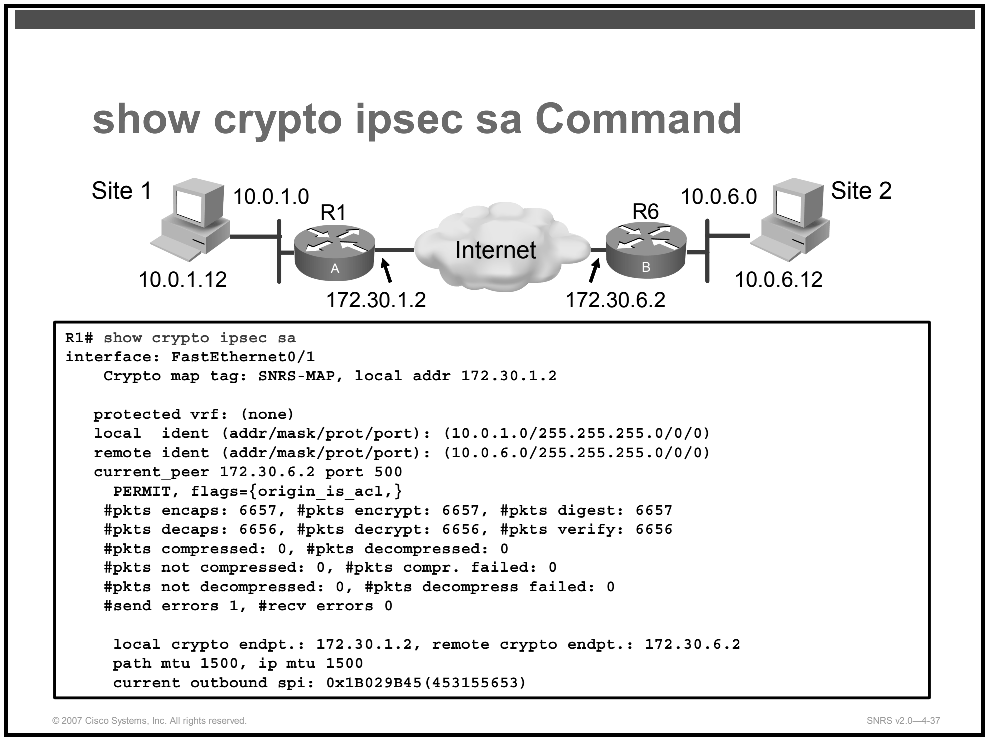
Use the **show crypto isakmp sa** command to view the state of current IKE SAs.

```
router# show crypto isakmp sa [ detail | nat | vrf ]
```

Syntax Description

detail	(Optional) Displays all existing IKE SAs, whether in an active or standby state
nat	(Optional) Displays IKE SAs that have undergone NAT
vrf	(Optional) Displays IKE SAs per VRF

Current SAs for the configured router will be shown. Use the **nat** keyword to display the IP address and port address of a remote peer when NAT is used.



show crypto ipsec sa Command

```
R1# show crypto ipsec sa
interface: FastEthernet0/1
    Crypto map tag: SNRS-MAP, local addr 172.30.1.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (10.0.6.0/255.255.255.0/0/0)
   current_peer 172.30.6.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 6657, #pkts encrypt: 6657, #pkts digest: 6657
    #pkts decaps: 6656, #pkts decrypt: 6656, #pkts verify: 6656
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 172.30.1.2, remote crypto endpt.: 172.30.6.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x1B029B45(453155653)
```

SNRS v2.0—4-37

Use the **show crypto ipsec sa** command to view the settings used by current SAs. If no keyword is used, all SAs are displayed.

```
router# show crypto ipsec sa [map map-name | address | identity |
interface interface-type interface-number | peer [vrf fvrf-name] address |
vrf ivrf-name | ipv6 [interface-type interface-number]] [detail]
```

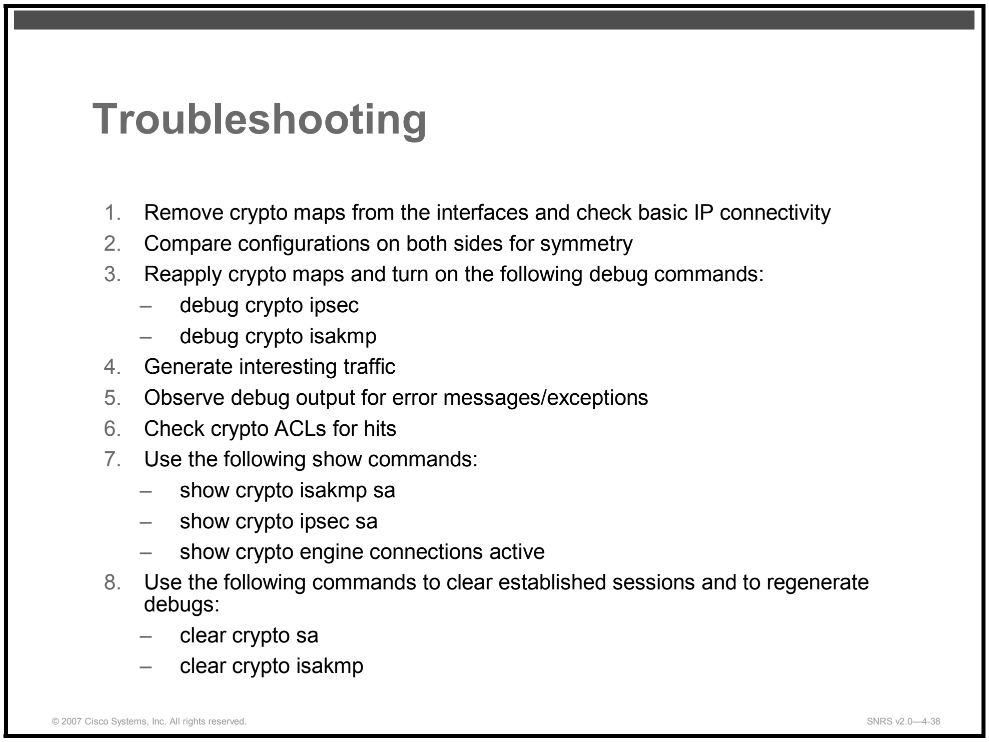
Syntax Description

map *map-name*	(Optional) Any existing SAs that were created for the crypto map set named *map-name* are displayed.
address	(Optional) All existing SAs are displayed, sorted by the destination address (either the local address or the address of the IPsec remote peer) and then by protocol (AH or ESP).
identity	(Optional) Only the flow information is displayed. It does not show the SA information.
interface *interface-type interface-number*	(Optional) All existing SAs created for an interface that is named *interface* are displayed.
peer [**vrf** *fvrf-name*] **address**	(Optional) All existing SAs with the peer address are displayed. If the peer address is in the VRF, specify **vrf** and *fvrf-name*.
vrf *ivrf-name*	(Optional) All existing SAs whose inside VRF (IVRF)is the same as the *ivrf-name* are displayed.
ipv6	(Optional) IPv6 crypto IPsec SAs are displayed.
detail	(Optional) Detailed error counters are displayed. (The default is the high-level send or receive error counters.)

If no keyword is used, all SAs are displayed. They are sorted first by interface, and then by traffic flow (for example, source or destination address, mask, protocol, or port). Within a flow, the SAs are listed by protocol (ESP or AH) and direction (inbound or outbound).

Troubleshooting

This topic describes some strategies for troubleshooting IPsec.



Troubleshooting

1. Remove crypto maps from the interfaces and check basic IP connectivity
2. Compare configurations on both sides for symmetry
3. Reapply crypto maps and turn on the following debug commands:
   - debug crypto ipsec
   - debug crypto isakmp
4. Generate interesting traffic
5. Observe debug output for error messages/exceptions
6. Check crypto ACLs for hits
7. Use the following show commands:
   - show crypto isakmp sa
   - show crypto ipsec sa
   - show crypto engine connections active
8. Use the following commands to clear established sessions and to regenerate debugs:
   - clear crypto sa
   - clear crypto isakmp

SNRS v2.0—4-38

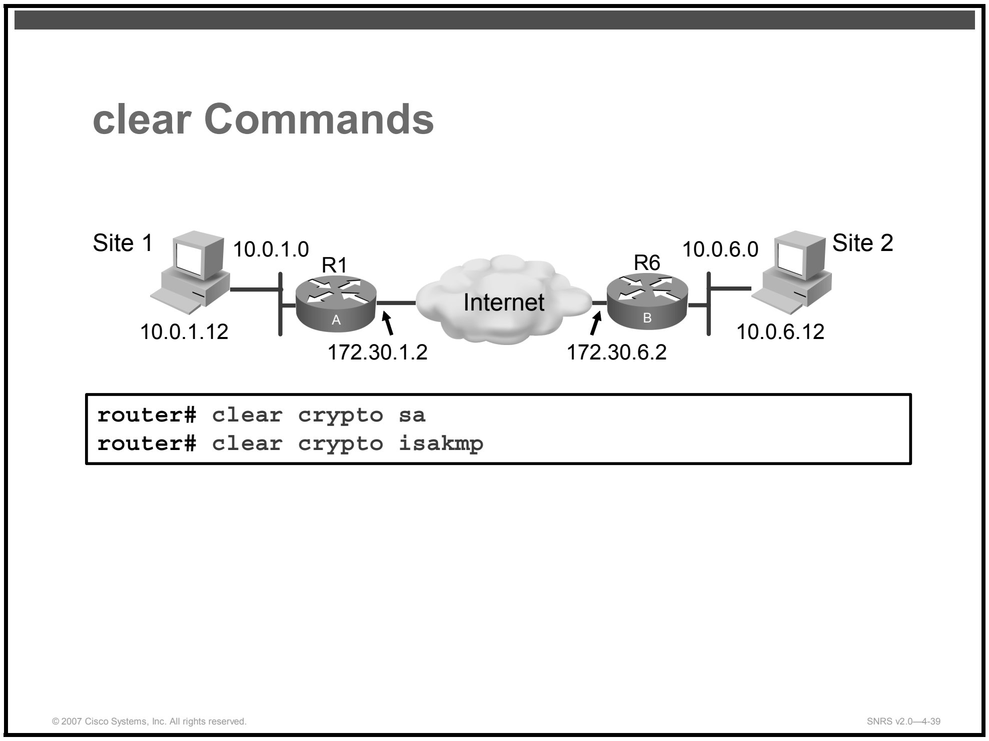
A good troubleshooting procedure is useful when verifying VPN operation.

When troubleshooting an IPsec problem, the following strategy may be employed:

1. Remove crypto maps from the interface and check for basic IP connectivity with a ping
2. Compare configurations on both sides of the connection
3. Reapply crypto maps and turn on debugging
4. Generate some traffic
5. Observer debug output for error messages
6. Check crypto ACLs for hits
7. Display ISAKMP and IPsec SAs
8. Clear current SAs to regenerate new debug messages after making changes

Using clear Commands

This section covers some **clear** commands used in troubleshooting.



clear Commands

```
router# clear crypto sa
router# clear crypto isakmp
```

When troubleshooting IPsec, you will need to clear some or all of the current SAs to apply changes or verify operation.

Two useful commands are the **clear crypto sa** and **clear crypto isakmp** commands.

Clearing IPsec SAs

Use the **clear crypto sa** command to delete IPsec SAs.

```
router# clear crypto sa [active | standby] [map map-name |
peer [vrf fvrf-name] address | entry destination-address
protocol spi | counters]
```

Syntax Description

active	(Optional) Clears only IPSec SAs that are in the active state.
standby	(Optional) Clears only IPSec SAs that are in the standby state. **Note:** If the router is in standby mode, the router will immediately resynchronize the standby SAs; thus, it may appear as though the standby SAs were not cleared.
peer [**vrf** *fvrf-name*] **address**	Deletes any IPsec SAs for the specified peer The *fvrf-name* argument specifies the front door VRF (FVRF) of the peer address.
vrf *ivrf-name*	(Optional) Clears all IPsec SAs whose IVRF is the same as the *ivrf-name*

`map`	Deletes any IPsec SAs for the named crypto map set
`map-name`	Specifies the name of a crypto map set
`entry`	Deletes the IPsec SA with the specified address, protocol, and security parameter index (SPI)
`destination-address`	Specifies the IP address of the remote peer.
`protocol`	Specifies either the ESP or AH
`spi`	Specifies an SPI (found by displaying the SA database)
`counters`	Clears the traffic counters maintained for each SA The **counters** keyword does not clear the SAs themselves.

This command clears (deletes) IPsec SAs.

If the SAs were established via IKE, they are deleted and future IPsec traffic will require new SAs to be negotiated. (When IKE is used, the IPsec SAs are established only when needed.)

If the SAs are manually established, the SAs are deleted and reinstalled. (When IKE is not used, the IPsec SAs are created as soon as the configuration is completed.)

Note If the **peer**, **map**, **entry**, **counters, active**, or **standby** keywords are not used, all IPsec SAs will be deleted.

If any of the above commands cause a particular SA to be deleted, all of the "sibling" SAs (that were established during the same IKE negotiation) are deleted as well.

The **counters** keyword simply clears the traffic counters maintained for each SA; it does not clear the SAs themselves.

If you make configuration changes that affect SAs, these changes will not apply to existing SAs but to negotiations for subsequent SAs. You can use the **clear crypto sa** command to restart all SAs so that they will use the most current configuration settings. In the case of manually established SAs, if you make changes that affect SAs, you must use the **clear crypto sa** command before the changes take effect.

If the router is processing active IPsec traffic, it is suggested that you clear only the portion of the SA database that is affected by the changes, to avoid causing active IPsec traffic to temporarily fail.

Note This command clears only IPsec SAs; to clear the IKE state, use the **clear crypto isakmp** command.

Clearing IKE SAs

Use the **clear crypto isakmp** command to clear active IKE connections.

```
Router# clear crypto isakmp [connection-id]
```

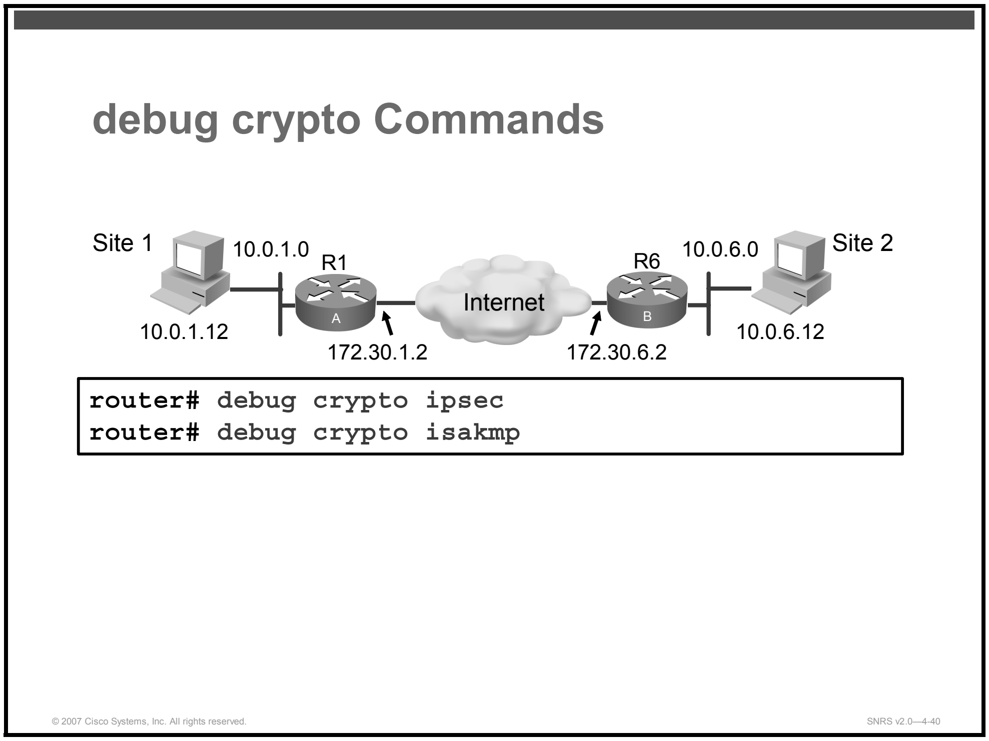
Syntax Description

connection-id	(Optional) ID of the connection that is to be cleared If this argument is not used, all existing connections will be cleared.

If the *connection-id* argument is not used, all existing IKE connections will be cleared when this command is issued.

Enabling debug Output for IPsec Events

This section describes how to enable debugging for IPsec traffic.



debug crypto Commands

```
router# debug crypto ipsec
router# debug crypto isakmp
```

SNRS v2.0—4-40

Various **debug** commands are available to use in troubleshooting IPsec operation.

Use the **debug crypto ipsec** and the **debug crypto isakmp** commands to display IPsec and ISAKMP events. The **no** form of these commands disables debugging output.

Caution Because these commands generate a significant amount of output for every IP packet processed, use them only when traffic on the IP network is low, so that other activity on the system is not adversely affected.

Displaying IPsec Events

To display messages about IPsec events, use the **debug crypto ipsec** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

The following is sample output from the **debug crypto ipsec** command. In this example, SAs have been successfully established.

```
router# debug crypto ipsec
router# clear crypto sa
router# ping
Protocol [ip]:
Target IP address: 10.0.6.2
Repeat count [5]:
```

```
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.0.1.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
```

Response

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.6.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
        Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
router# show logging
*Sep 29 18:15:43.643: IPSEC(sa_request): ,
  (key eng. msg.) OUTBOUND local= 172.30.1.2, remote= 172.30.6.2,
    local_proxy= 10.0.1.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 10.0.6.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= NONE  (Tunnel),
    lifedur= 3600s and 4608000kb,
    spi= 0xADAF963F(2913965631), conn_id= 0, keysize= 0, flags= 0x0
*Sep 29 18:15:43.651: IPSEC(validate_proposal_request): proposal part #1
*Sep 29 18:15:43.651: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.30.1.2, remote= 172.30.6.2,
    local_proxy= 10.0.1.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 10.0.6.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des  (Tunnel),
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0
*Sep 29 18:15:43.651: Crypto mapdb : proxy_match
        src addr     : 10.0.1.0
        dst addr     : 10.0.6.0
        protocol     : 0
        src port     : 0
*Sep 29 18:15:43.655: IPSEC(key_engine): got a queue event with 1 KMI
message(s)
```

```
*Sep 29 18:15:43.655: Crypto mapdb : proxy_match
        src addr     : 10.0.1.0
        dst addr     : 10.0.6.0
        protocol     : 0
        src port     : 0
        dst port     : 0
*Sep 29 18:15:43.655: IPSEC(crypto_ipsec_sa_find_ident_head): reconnecting
with the same proxies and peer 172.30.6.2
*Sep 29 18:15:43.655: IPSEC(policy_db_add_ident): src 10.0.1.0, dest 10.0.6.0,
dest_port 0
*Sep 29 18:15:43.655: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.30.1.2, sa_proto= 50,
    sa_spi= 0xADAF963F(2913965631),
    sa_trans= esp-des , sa_conn_id= 2015
*Sep 29 18:15:43.655: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.30.6.2, sa_proto= 50,
    sa_spi= 0xBD872F62(3179753314),
    sa_trans= esp-des , sa_conn_id= 2016
*Sep 29 18:15:43.655: IPSEC(update_current_outbound_sa): updated peer
172.30.6.2 current outbound sa to SPI BD872F62
```

Displaying ISAKMP Events

To display messages about IKE events, use the **debug crypto isakmp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
router# debug crypto isakmp
router# clear crypto isakmp
router# ping
Protocol [ip]:
Target IP address: 10.0.6.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.0.1.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
```

Response

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.6.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
router# show logging
```

Key exchange starts:

```
(1009):Old State = IKE_I_MM4  New State = IKE_I_MM5
*Sep 29 18:30:26.023: ISAKMP (0:1009): received packet from 172.30.6.2 dport
500 sport 500 Global (I) MM_KEY_EXCH
*Sep 29 18:30:26.023: ISAKMP:(1009): processing ID payload. message ID = 0
*Sep 29 18:30:26.023: ISAKMP (0:1009): ID payload
        next-payload : 8
        type         : 1
        address      : 172.30.6.2
        protocol     : 17
        port         : 500
        length       : 12
*Sep 29 18:30:26.023: ISAKMP:(0):: peer matches *none* of the profiles
*Sep 29 18:30:26.023: ISAKMP:(1009): processing HASH payload. message ID = 0
```

Authentication successful!

```
*Sep 29 18:30:26.023: ISAKMP:(1009):SA authentication status:
authenticated
*Sep 29 18:30:26.023: ISAKMP:(1009):SA has been authenticated with 172.30.6.2
*Sep 29 18:30:26.023: ISAKMP: Trying to insert a peer
172.30.1.2/172.30.6.2/500/,  and inserted successfully 6437A900.
*Sep 29 18:30:26.027: ISAKMP:(1009):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Sep 29 18:30:26.027: ISAKMP:(1009):Old State = IKE_I_MM5  New State =
IKE_I_MM6
*Sep 29 18:30:26.027: ISAKMP:(1009):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_MAIN_MODE
*Sep 29 18:30:26.027: ISAKMP:(1009):Old State = IKE_I_MM6  New State =
IKE_I_MM6
*Sep 29 18:30:26.027: ISAKMP:(1009):Input = IKE_MESG_INTERNAL,
IKE_PROCESS_COMPLETE
```

IKE Phase 1 complete!

```
*Sep 29 18:30:26.027: ISAKMP:(1009):Old State = IKE_I_MM6  New State =
IKE_P1_COMPLETE
*Sep 29 18:30:26.027: ISAKMP:(1009):beginning Quick Mode exchange, M-ID of -
1804995995
```

```
*Sep 29 18:30:26.027: ISAKMP:(1009):QM Initiator gets spi

*Sep 29 18:30:26.031: ISAKMP:(1009): sending packet to 172.30.6.2 my_port 500
peer_port 500 (I) QM_IDLE

*Sep 29 18:30:26.031: ISAKMP:(1009):Node -1804995995, Input =
IKE_MESG_INTERNAL, IKE_INIT_QM

*Sep 29 18:30:26.031: ISAKMP:(1009):Old State = IKE_QM_READY  New State =
IKE_QM_I_QM1

*Sep 29 18:30:26.031: ISAKMP:(1009):Input = IKE_MESG_INTERNAL,
IKE_PHASE1_COMPLETE

*Sep 29 18:30:26.031: ISAKMP:(1009):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Sep 29 18:30:26.035: ISAKMP (0:1009): received packet from 172.30.6.2 dport
500 sport 500 Global (I) QM_IDLE

*Sep 29 18:30:26.035: ISAKMP:(1009): processing HASH payload. message ID = -
1804995995

*Sep 29 18:30:26.035: ISAKMP:(1009): processing SA payload. message ID = -
1804995995
```

Checking IPsec attributes proposed by peer:

```
*Sep 29 18:30:26.035: ISAKMP:(1009):Checking IPSec proposal 1

*Sep 29 18:30:26.035: ISAKMP: transform 1, ESP_DES

*Sep 29 18:30:26.035: ISAKMP:   attributes in transform:

*Sep 29 18:30:26.035: ISAKMP:      encaps is 1 (Tunnel)

*Sep 29 18:30:26.035: ISAKMP:      SA life type in seconds

*Sep 29 18:30:26.035: ISAKMP:      SA life duration (basic) of 3600

*Sep 29 18:30:26.035: ISAKMP:      SA life type in kilobytes

*Sep 29 18:30:26.035: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50
0x0
```

Attributes are accepted!

```
*Sep 29 18:30:26.035: ISAKMP:(1009):atts are acceptable.

*Sep 29 18:30:26.039: ISAKMP:(1009): processing NONCE payload. message ID = -
1804995995

*Sep 29 18:30:26.039: ISAKMP:(1009): processing ID payload. message ID = -
1804995995

*Sep 29 18:30:26.039: ISAKMP:(1009): processing ID payload. message ID = -
1804995995

*Sep 29 18:30:26.039: ISAKMP:(1009): Creating IPSec SAs

*Sep 29 18:30:26.039:         inbound SA from 172.30.6.2 to 172.30.1.2 (f/i)
0/ 0        (proxy 10.0.6.0 to 10.0.1.0)

*Sep 29 18:30:26.039:         has spi 0xD828DCD9 and conn_id 0

*Sep 29 18:30:26.039:         lifetime of 3600 seconds

*Sep 29 18:30:26.039:         lifetime of 4608000 kilobytes

*Sep 29 18:30:26.039:         outbound SA from 172.30.1.2 to 172.30.6.2 (f/i)
0/0        (proxy 10.0.1.0 to 10.0.6.0)
```

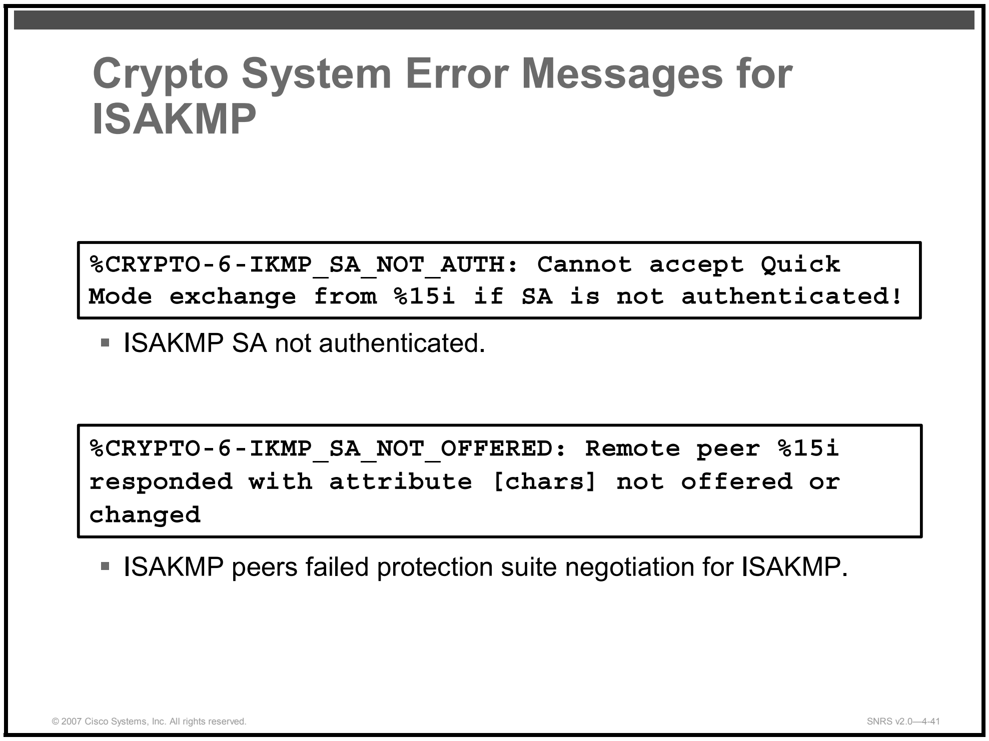
```
*Sep 29 18:30:26.039:         has spi  0x91BA79A9 and conn_id 0

*Sep 29 18:30:26.039:         lifetime of 3600 seconds

*Sep 29 18:30:26.039:         lifetime of 4608000 kilobytes

*Sep 29 18:30:26.039: ISAKMP:(1009): sending packet to 172.30.6.2 my_port 500
peer_port 500 (I) QM_IDLE

*Sep 29 18:30:26.039: ISAKMP:(1009):deleting node -1804995995 error FALSE
reason "No Error"

*Sep 29 18:30:26.039: ISAKMP:(1009):Node -1804995995, Input =
IKE_MESG_FROM_PEER, IKE_QM_EXCH
```

IKE Phase 2 complete!

```
*Sep 29 18:30:26.039: ISAKMP:(1009):Old State = IKE_QM_I_QM1  New State =
IKE_QM_PHASE2_COMPLETE
```

Examining Output for ISAKMP Event Errors

This topic describes some ISAKMP event errors.



Crypto System Error Messages for ISAKMP

```
%CRYPTO-6-IKMP_SA_NOT_AUTH: Cannot accept Quick
Mode exchange from %15i if SA is not authenticated!
```

- ISAKMP SA not authenticated.

```
%CRYPTO-6-IKMP_SA_NOT_OFFERED: Remote peer %15i
responded with attribute [chars] not offered or
changed
```

- ISAKMP peers failed protection suite negotiation for ISAKMP.

SNRS v2.0—4-41

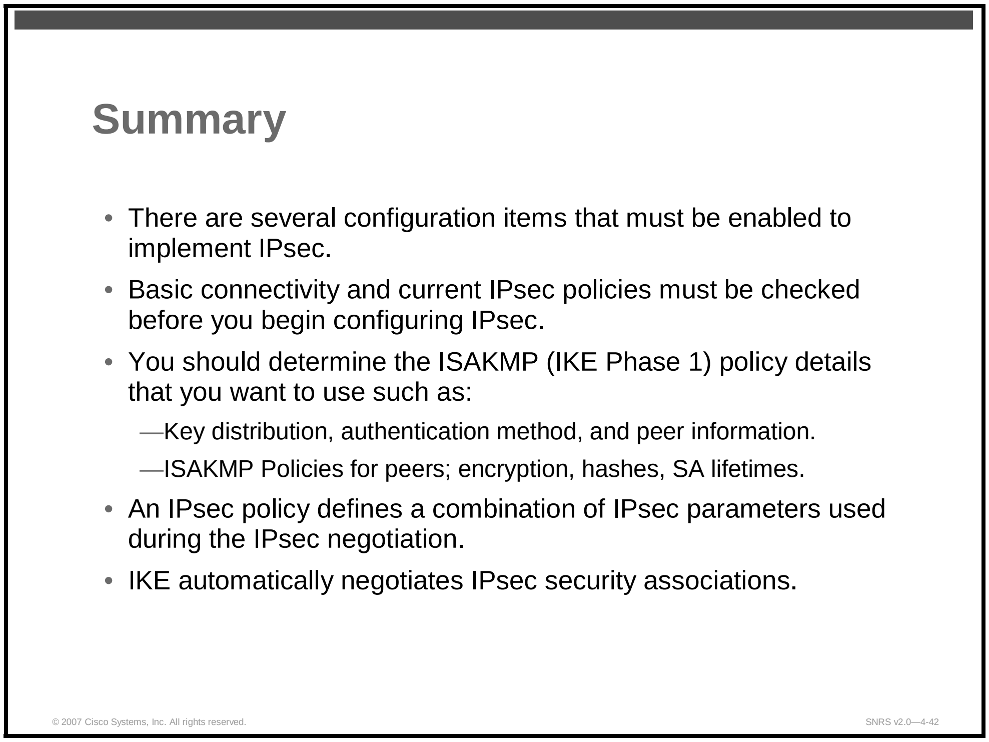
Cisco IOS Software can generate many useful system error messages for ISAKMP. Two of the error messages are as follows:

- **%CRYPTO-6-IKMP_SA_NOT_AUTH: Cannot accept Quick Mode exchange from %15i if SA is not authenticated!:** The ISAKMP SA with the remote peer was not authenticated, yet the peer attempted to begin a quick mode exchange. This exchange must only be done with an authenticated SA. The recommended action is to contact the remote peer administrator to resolve the improper configuration.

- **%CRYPTO-6-IKMP_SA_NOT_OFFERED: Remote peer %15i responded with attribute [chars] not offered or changed:** ISAKMP peers negotiate policy by the initiator offering a list of possible alternate protection suites. The responder replied with an ISAKMP policy that the initiator did not offer. The recommended action is to contact the remote peer administrator to resolve the improper configuration.

Summary

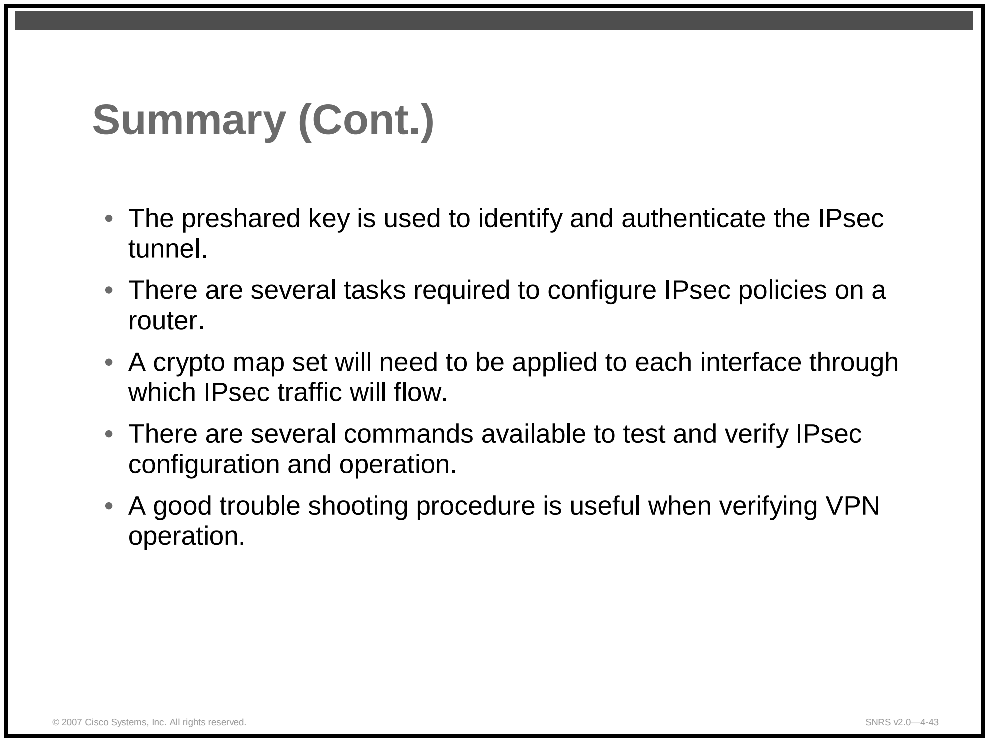
This topic summarizes the key points that were discussed in this lesson.



Summary

- There are several configuration items that must be enabled to implement IPsec.
- Basic connectivity and current IPsec policies must be checked before you begin configuring IPsec.
- You should determine the ISAKMP (IKE Phase 1) policy details that you want to use such as:
  - —Key distribution, authentication method, and peer information.
  - —ISAKMP Policies for peers; encryption, hashes, SA lifetimes.
- An IPsec policy defines a combination of IPsec parameters used during the IPsec negotiation.
- IKE automatically negotiates IPsec security associations.

SNRS v2.0—4-42



Summary (Cont.)

- The preshared key is used to identify and authenticate the IPsec tunnel.
- There are several tasks required to configure IPsec policies on a router.
- A crypto map set will need to be applied to each interface through which IPsec traffic will flow.
- There are several commands available to test and verify IPsec configuration and operation.
- A good trouble shooting procedure is useful when verifying VPN operation.

SNRS v2.0—4-43

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Deploying IPsec Virtual Private Networks*. http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns142/networking_solutions_white_paper09186a0080117919.shtml.
- Cisco Systems, Inc. *Configuring IPsec Network* Security.http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt4/scipsec.htm.
- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4*. http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_book09186a008043360a.html.

Lesson 4

Implementing IPsec VPNs Using PKI

Overview

IP Security (IPsec) virtual private networks (VPNs) can be configured for various types of authentication. One such method is using pre-shared keys. In that case, each client shares a common key. That method is not very scalable especially in an enterprise network. Another more scalable method would incorporate the public key infrastructure (PKI) for authentication purposes. This lesson guides you through the process of configuring an IPsec site-to-site VPN using PKI.

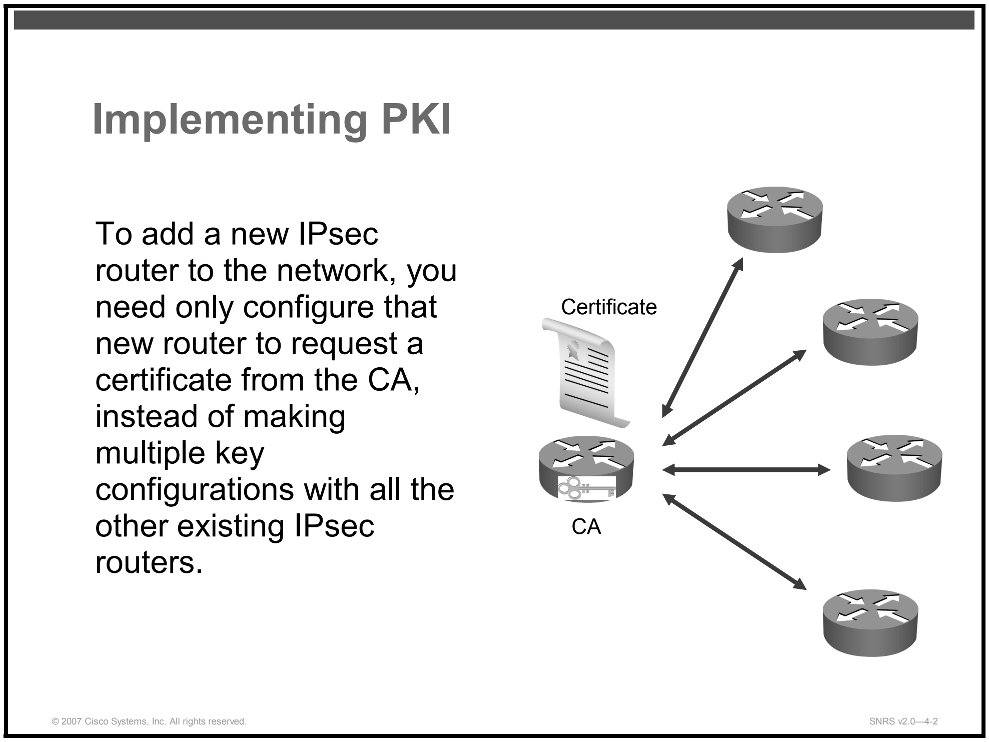
Objectives

Upon completing this lesson, you will be able to configure an IPsec site-to-site VPN using PKI. This ability includes being able to meet these objectives:

- Describe Cisco IOS PKI support
- Describe digital signatures
- Describe how SCEP manages the certificate life cycle
- Describe how to configure an IPsec site-to-site VPN using digital certificates on a Cisco router
- Describe the commands used to test and verify IPsec CA configurations

Examining Cisco IOS PKI

This topic describes Cisco IOS PKI support.



Implementing PKI

To add a new IPsec router to the network, you need only configure that new router to request a certificate from the CA, instead of making multiple key configurations with all the other existing IPsec routers.

SNRS v2.0—4-2

Cisco IOS PKI provides certificate management to support security protocols such as IPsec, Secure Shell (SSH), and Secure Sockets Layer (SSL).

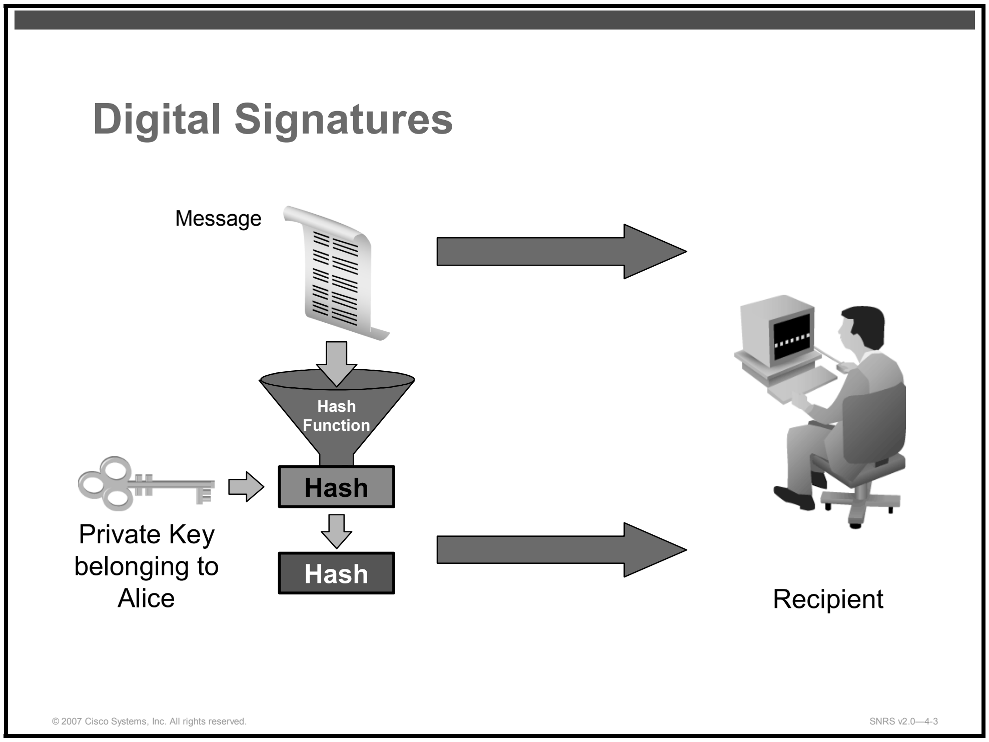
A PKI is composed of the following entities:

- Peers communicating on a secure network
- Digital signatures (Rivest, Shamir, and Adleman [RSA] keys)
- At least one certificate authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, the CA that granted the certificate and the digital signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption keys and identity information in a secured data network. Every device participating in the secured communication is enrolled in the PKI in a process where the device generates an RSA key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

Digital Signatures

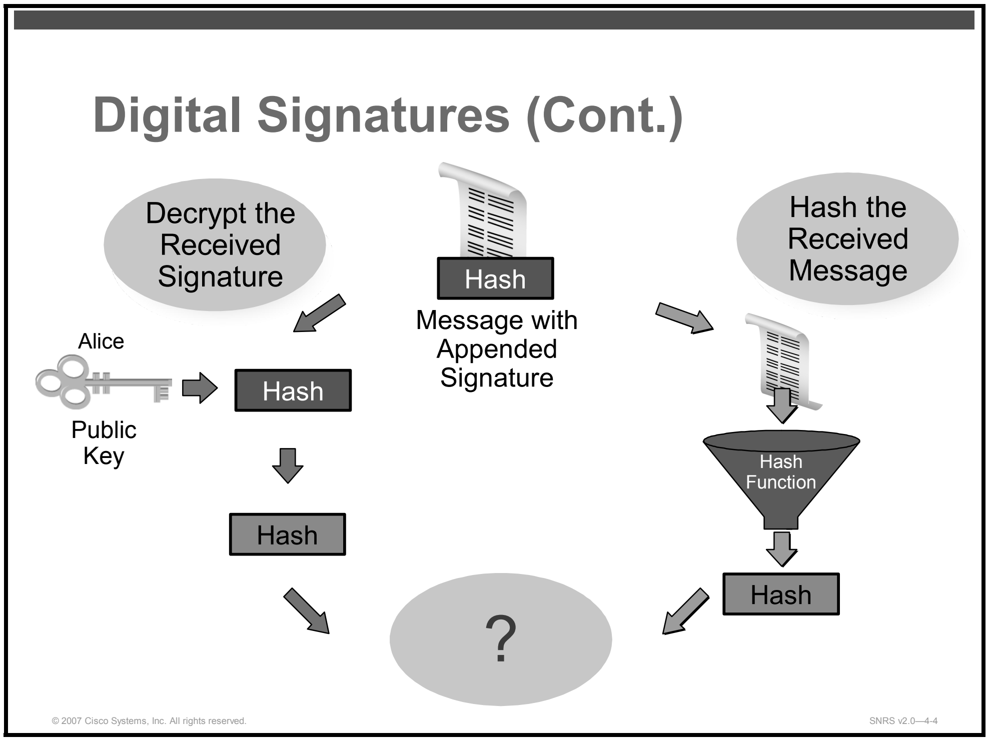
This topic described digital signatures.



Digital Signatures

SNRS v2.0—4-3

Digital signatures, enabled by public key cryptography, provide a means of digitally authenticating devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with the private key of a user.



Digital Signatures (Cont.)

Decrypt the Received Signature

Hash

Message with Appended Signature

Hash the Received Message

Alice

Public Key

Hash

Hash

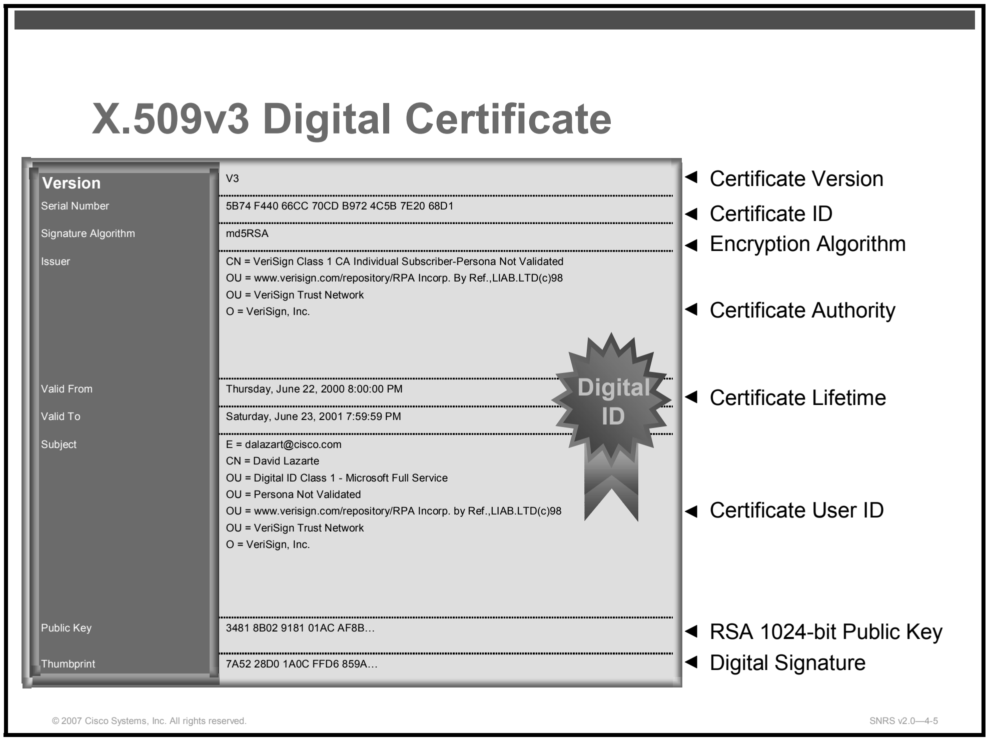
Hash Function

Hash

?

SNRS v2.0—4-4

The receiver verifies the signature by decrypting the message with the public key of the sender. The fact that the message could be decrypted using the public key of the sender indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver having a copy of the public key of the sender and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender. Internet Key Exchange (IKE), an essential component of IPsec, can use digital signatures to authenticate peer devices before setting up security associations (SAs).



X.509v3 Digital Certificate

Version	V3
Serial Number	5B74 F440 66CC 70CD B972 4C5B 7E20 68D1
Signature Algorithm	md5RSA

Issuer	CN = VeriSign Class 1 CA Individual Subscriber-Persona Not Validated
	OU = www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98
	OU = VeriSign Trust Network
	O = VeriSign, Inc.

Valid From	Thursday, June 22, 2000 8:00:00 PM
Valid To	Saturday, June 23, 2001 7:59:59 PM

Subject	E = dalazart@cisco.com
	CN = David Lazarte
	OU = Digital ID Class 1 - Microsoft Full Service
	OU = Persona Not Validated
	OU = www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)98
	OU = VeriSign Trust Network
	O = VeriSign, Inc.

Public Key	3481 8B02 9181 01AC AF8B…
Thumbprint	7A52 28D0 1A0C FFD6 859A…

◄ Certificate Version

◄ Certificate ID

◄ Encryption Algorithm

◄ Certificate Authority

◄ Certificate Lifetime

◄ Certificate User ID

◄ RSA 1024-bit Public Key

◄ Digital Signature

SNRS v2.0—4-5

Digital certificates provide the link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the public key of the entity. The certificate is itself signed by a CA, a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates. X.509 specifies the digital certificate format.

To validate the signature of the CA, the receiver must first know the public key of the CA. Normally, this process is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default.

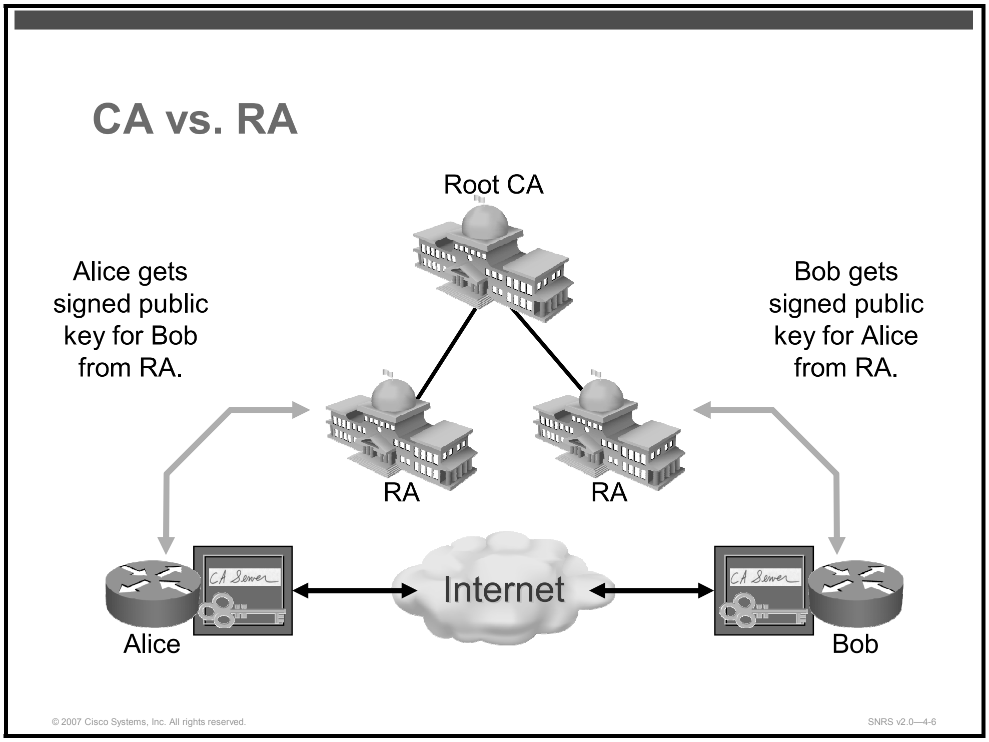
Without digital signatures, you must manually exchange either public keys or secrets between each pair of devices that use IPsec to protect communications between them. Without certificates, every new device added to the network requires a configuration change on every other device with which it communicates securely. With digital certificates, each device is enrolled with a CA. When two devices wish to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, someone simply enrolls that device with a CA, and none of the other devices needs modification. When the new device attempts an IPsec connection, certificates are automatically exchanged, and the device can be authenticated. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

RSA Keys

RSA key pairs are used to sign and encrypt IKE key management messages and are required before you can obtain a certificate for your router.

Certificate Authorities

This section describes the use of CAs and RAs.



CA vs. RA

Alice gets signed public key for Bob from RA.

Bob gets signed public key for Alice from RA.

SNRS v2.0—4-6

Using CAs simplifies the administration of IPsec network devices. You can use a CA with a network containing multiple IPsec-compliant devices such as routers.

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use a CA provided by a third-party CA vendor, or you can use an internal CA, which is the Cisco IOS certificate server.

CA vs. RA

Because PKIs are hierarchical in nature, the issuing CA may be a root CA (the top-level CA in the hierarchy), or a subordinate CA. The root CA uses a self-signed certificate, and the subordinate CA certificate is signed by the CA above it. The PKI might employ additional hosts, RAs, to accept requests for enrollment in the PKI. RAs are employed to reduce the burden on CAs in an environment that supports a large number of certificate transactions.

The CA is the central point of trust within the PKI. The end hosts in the organization trust the CA as the decisive source of information for the authenticity of other end hosts. When the CA issues a certificate, its digital signature on the certificate is a definitive mark that the end host, which holds the certificate, is part of the PKI.

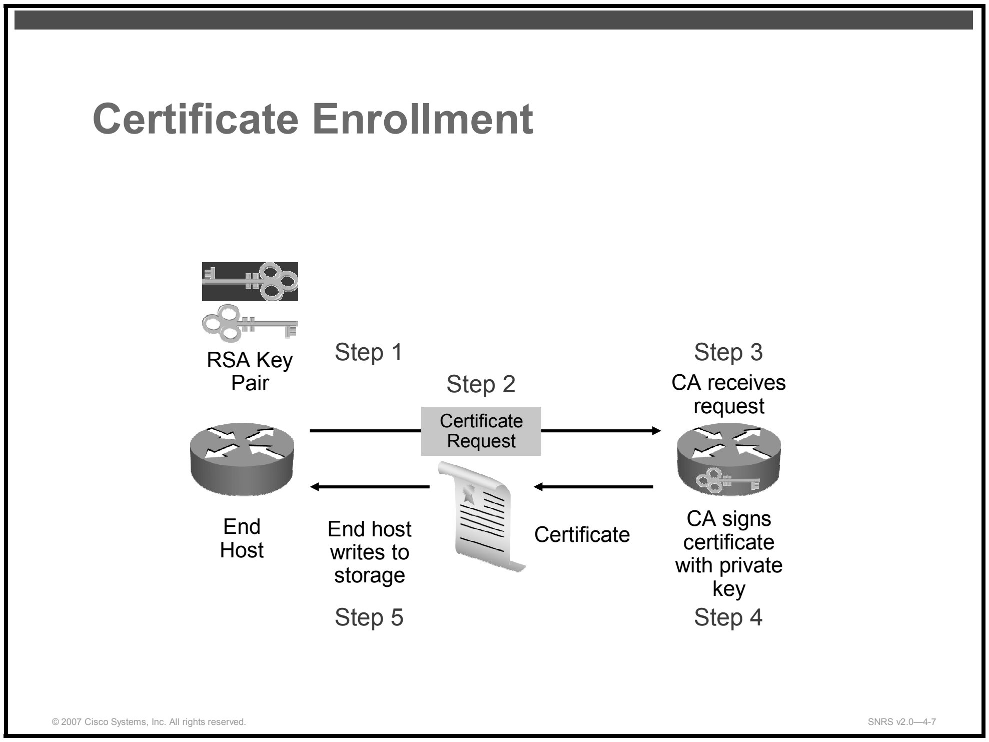
In a more complex environment, the RA might be tasked with verifying user identity, establishing passwords for certificate management transactions, submitting enrollment requests (along with appropriate organizational attributes or other information) to the CA, and handling assorted tasks (for example, certificate revocation and re-enrollment).

The RA only has the power to accept registration requests and forward them to the CA. The RA is not allowed to issue certificates or publish CRLs. The CA is responsible for these functions.

PKI and Accurate Time

An accurate time source must be available to enroll a cryptographic device in a PKI and to check certificate validity from negotiating peers.

When crypto peers present their certificate to each other, the validity date is among the first things that will be checked within the certificate. Cisco IOS Software will compare the beginning and end of the certificate validity period (embedded in the certificate) to the time and date in the clock of the router. If the current date of the router is within the validity period of the certificate, the router goes on to check the validity of other components of the certificate. The router must have access to the correct time, either through manual configuration of the system clock, accurate time sources such as Network Time Protocol (NTP), or clock adjustment via Simple Network Management Protocol (SNMP).



Certificate Enrollment

RSA Key Pair

Step 1

Step 2

Certificate Request

Step 3

CA receives request

End Host

End host writes to storage

Step 5

Certificate

CA signs certificate with private key

Step 4

SNRS v2.0—4-7

Certificate Enrollment

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA.

The following steps describe the certificate enrollment process:

1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request and, depending on your network configuration, one of the following options occurs:
   - Manual intervention is required to approve the request.
   - The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time that the enrollment request is sent to the CA server.
4. After the request is approved, the CA signs the certificate with its private key and returns the completed certificate to the end host.
5. The end host writes the certificate to a storage area such as NVRAM.

Supported Certificate Enrollment Methods

Cisco IOS Software supports the following methods to obtain a certificate from a CA:

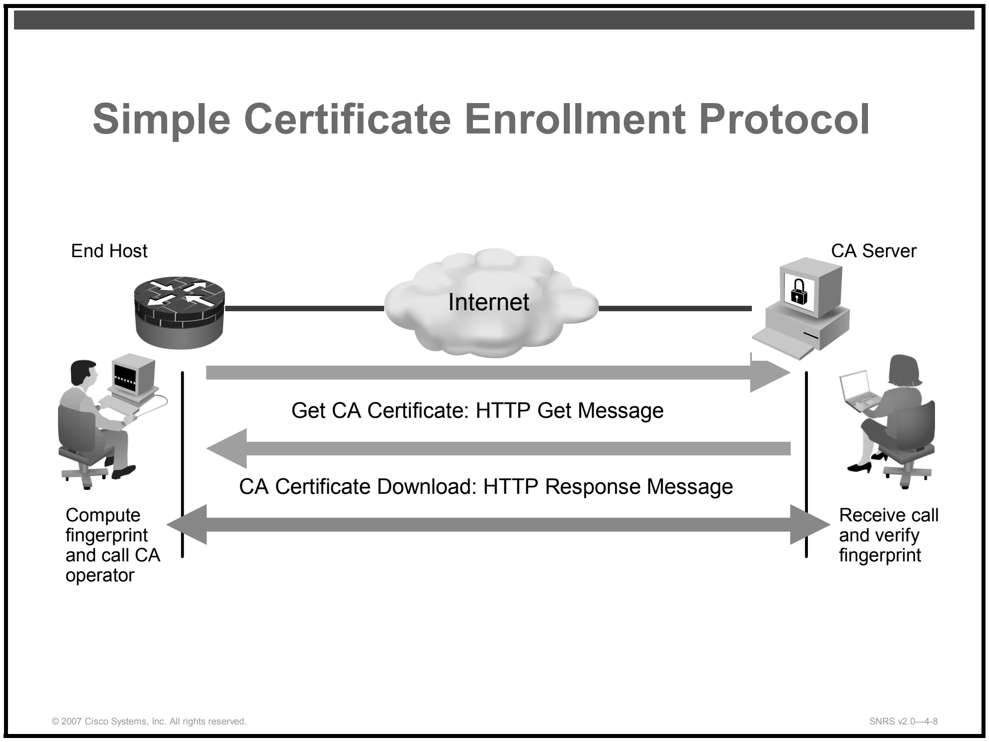
- **Simple Certificate Enrollment Protocol (SCEP):** SCEP is a Cisco developed enrollment protocol that uses HTTP to communicate with the CA or RA. SCEP is the most commonly used method for sending and receiving requests and certificates.

Note To take advantage of automated certificate and key rollover functionality, Cisco IOS Release 12.4(2)T must be running and SCEP must be used as your client enrollment method.

- **Public-Key Cryptography Standard (PKCS) #12:** The router imports certificates in PKCS #12 format from an external server.
- **Cisco IOS File System (IFS):** The router uses any file system that is supported by Cisco IOS Software (such as TFTP, FTP, flash memory, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable Cisco IFS certificate enrollment when their CA does not support SCEP.
- **Manual (cut and paste):** The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the terminal. Users may manually cut and paste certificate requests and certificates when they do not have a network connection between the router and CA.
- **Enrollment profiles:** The router sends HTTP-based enrollment requests directly to the CA server instead of the RA proxy. Enrollment profiles can be used if a CA server does not support SCEP and if the user does not want to use an RA as a proxy.
- **Self-signed certificate enrollment for a trustpoint:** The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the SSL handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time that the router reloaded.

Examining SCEP

This topic describes how SCEP manages the certificate life cycle.



Simple Certificate Enrollment Protocol

End Host

CA Server

Internet

Get CA Certificate: HTTP Get Message

CA Certificate Download: HTTP Response Message

Compute fingerprint and call CA operator

Receive call and verify fingerprint

SNRS v2.0—4-8

Cisco IOS Software uses SCEP to communicate with a PKI. Cisco Systems developed SCEP to extend the capability of the certificate enrollment protocol that was developed by VeriSign for Cisco. SCEP has achieved broad acceptance with the majority of CA software manufacturers, and Cisco competitors frequently implement protocols for certificate enrollment on their own VPN products.

SCEP offers a mechanism to support the secure transportation of key information and certificates between the different components of a PKI.

SCEP has these features:

- Is a transaction-oriented request and response protocol
- Uses Public-Key Cryptography Standard #7 (PKCS #7) (Cryptographic Message Syntax Standard) and Public-Key Cryptography Standard #10 (PKCS #10) (Certification Request Syntax Standard) to make requests from the CA server
- Is transport mechanism independent
- Requires manual authentication during enrollment

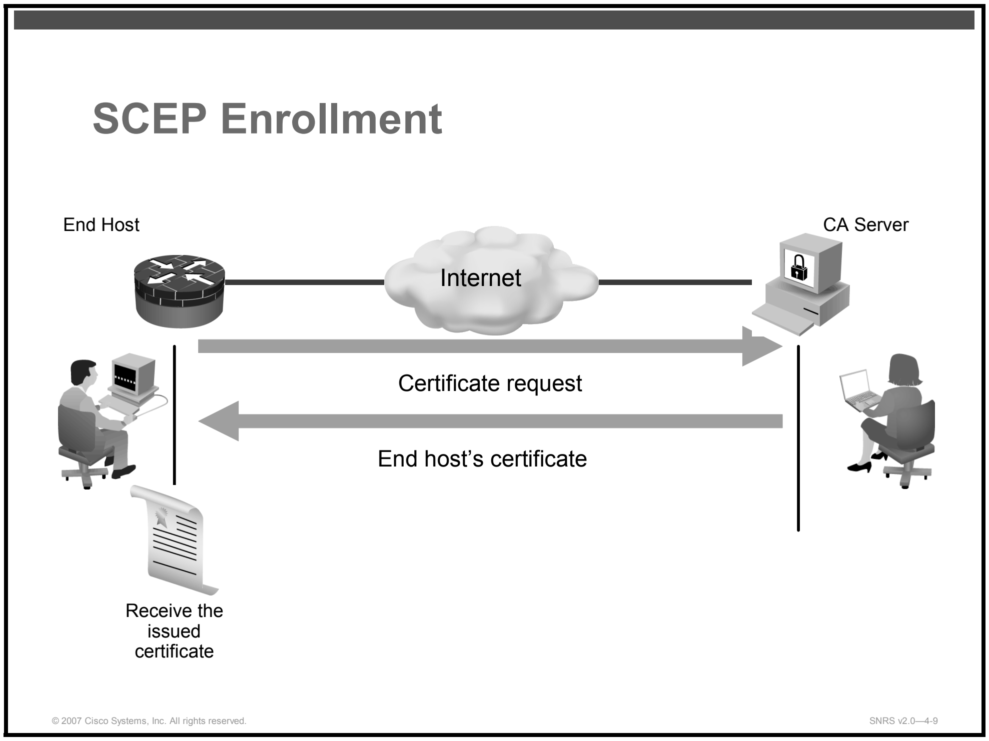
Operations supported by SCEP include:

- CA and RA public key distribution
- Certificate enrollment
- Certificate revocation
- Certificate query
- CRL query

SCEP employs the HTTP transport. Therefore, there is no requirement to implement support for new protocols on existing networks in the event that firewalls must be configured to permit access to services on protected networks.

The end hosts employ a standard format for transportation of certificates and key information when they communicate internally and with the CA and RA. The 15 PKCSs define these. RSA Security, a leading authority in development and maintenance of public key cryptography technology, publishes the PKCS documentation. The two standards that are applied with SCEP are PKCS #7 and PKCS #10.

As shown in the figure, CA and RA public key distribution is performed as cleartext HTTP transfers. After the end host receives the CA certificate, it has to authenticate the CA certificate by comparing the fingerprint (digital signature) against the fingerprint known at the CA or RA. This comparison is generally performed out of band, via e-mail or phone conversation between the remote user enrolling the end host and the operator at the CA or RA console.



SCEP Enrollment

End Host

CA Server

Internet

Certificate request

End host's certificate

Receive the issued certificate

SNRS v2.0—4-9

The end host may be enrolled via SCEP after retrieving the public key from the CA or RA. The enrollment request consists of a PKCS #10-formatted certificate request that is transmitted to the CA or RA in a PKCS #7 package. The certificate request will also include a challenge password and a request to include additional information in the certificate that the CA will return. The end host operator or administrator should know the challenge password, which is provided as an out-of-band authentication method for certificate issuance and verification activities. It will be made available to the CA or RA operator after the CA or RA receives the certificate request. As the end host generates the certificate request, the public key of the CA or RA public key signs it.

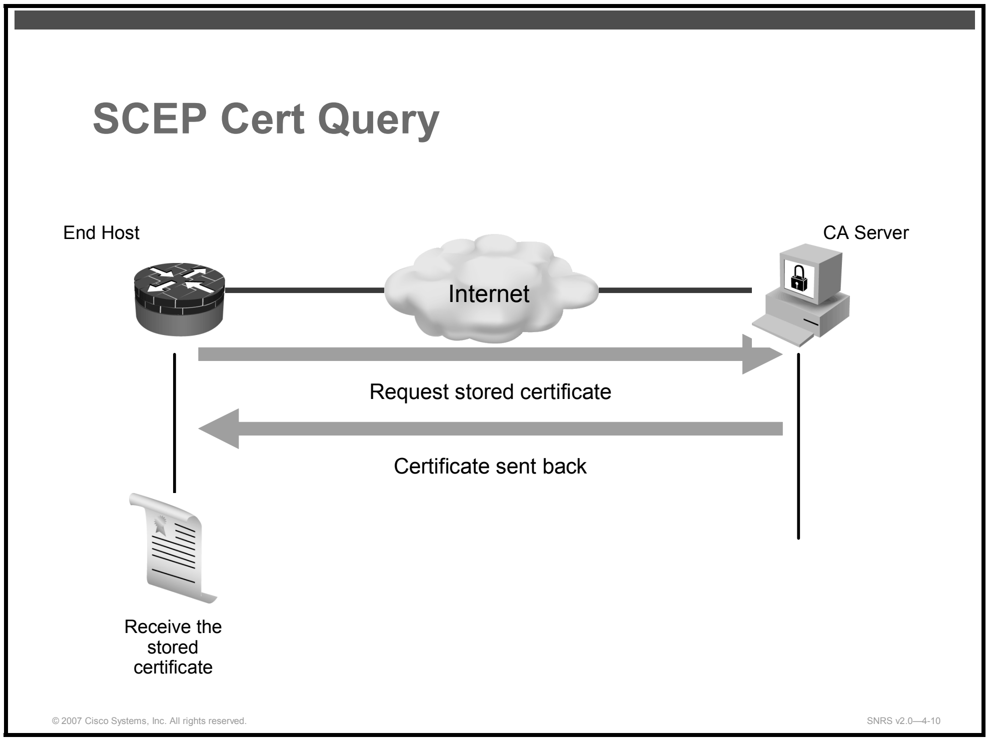
The CA or RA decrypts the certificate request with its private key and takes one of these two actions:

1. Automatically signs the certificate with its private key and returns the certificate to the end host

Or

2. Waits until the CA or RA operator verifies the certificate request with the end host operator and approves the request, after which the CA or RA signs the certificate and returns the certificate to the end host.

SCEP supports certificate revocation by an out-of-band dialog between the end host operator and the CA or RA operator. In the event that the keys of the end host are compromised, or if circumstances render the certificate invalid, the end host operator will contact the CA or RA operator and present the challenge password. This is known at the end host and at the CA or RA, because it was sent with the enrollment request. After the challenge password is verified, the CA or RA operator will follow the procedure for the given CA to revoke the certificate of the end host, and the revoked certificate will be published to the CRL.



SCEP Cert Query

End Host

CA Server

Internet

Request stored certificate

Certificate sent back

Receive the stored certificate

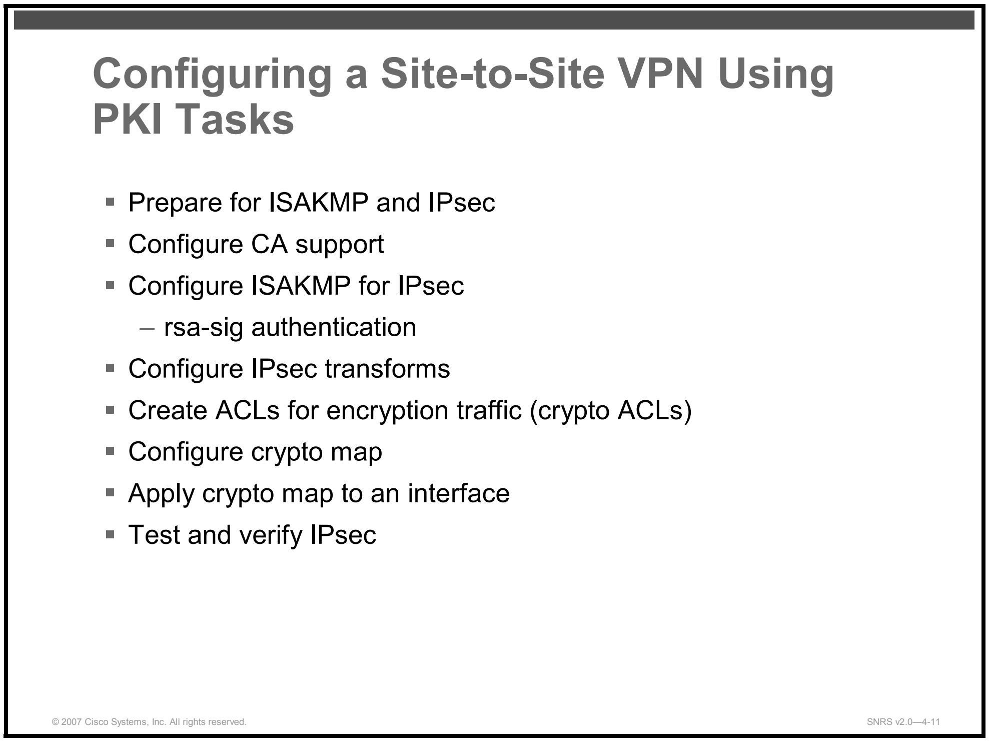
SNRS v2.0—4-10

As shown in the figure, if the end host does not have adequate memory to store its certificate, it may use the certificate query capability of SCEP to retrieve its certificate from the CA. Alternatively, the certificate query may be completed via LDAP. In either case, the end host must know the serial number of the certificate and the fully qualified domain name used in the certificate enrollment request.

The last functionality that SCEP supports is CRL checking. When an end host is presented with a certificate, it will extract the URL for the CRL distribution point from the certificate, and try to check if the presented certificate is listed on the CRL. With Cisco IOS Software, SCEP is the lowest of three preferences for the CRL checking protocol. HTTP is the most highly preferred option, followed by LDAP.

Configuring IPsec VPN Using Digital Certificates

This topic describes how to configure an IPsec site-to-site VPN using digital certificates on a Cisco router.



Configuring a Site-to-Site VPN Using PKI Tasks

- Prepare for ISAKMP and IPsec
- Configure CA support
- Configure ISAKMP for IPsec
  - rsa-sig authentication
- Configure IPsec transforms
- Create ACLs for encryption traffic (crypto ACLs)
- Configure crypto map
- Apply crypto map to an interface
- Test and verify IPsec

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-11

The configuration process for a site-to-site IPsec VPN using digital certificates consists of these five major tasks:

1. Prepare for Internet Security Association and Key Management Protocol (ISAKMP):

Preparing for ISAKMP and IPsec involves determining the detailed encryption policy: identifying the hosts and networks that you wish to protect, determining IPsec peer details, determining the IPsec features that you need, and ensuring that existing access control lists (ACLs) are compatible with IPsec.

Note You will prepare for ISAKMP and IPsec just as you did in the "Implementing IPsec VPNs Using Pre-Shared Keys" lesson on site-to-site VPNs using pre-shared keys.

2. Configure CA support.

This task involves setting the router clock, hostname, and domain name, generating the RSA keys, declaring a CA, and authenticating the CA and requesting your own certificate.

3. Configure ISAKMP for IPsec.

Configuring ISAKMP involves enabling ISAKMP, creating the ISAKMP policies, and validating the configuration.

Note You configure ISAKMP in the same manner as you did with pre-shared keys, except that you will be configuring the ISAKMP policy to authenticate using RSA signatures.

4. Configure IPsec transform sets.

IPsec configuration includes defining the transform sets, creating crypto ACLs, creating crypto map entries, and applying crypto map sets to interfaces.

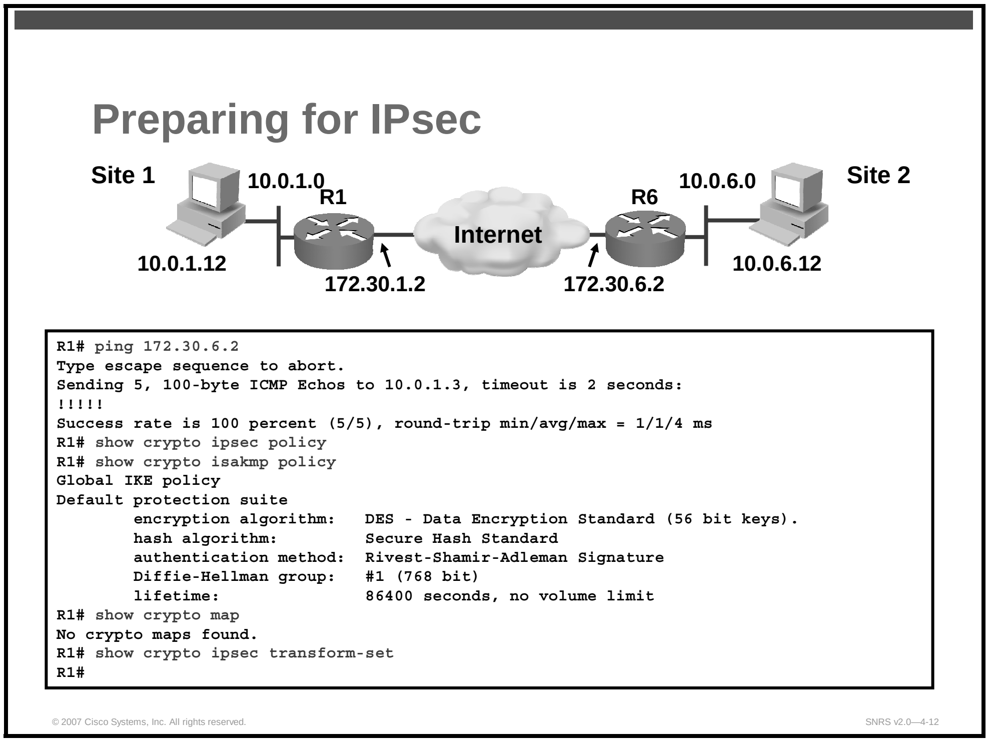
Note You will configure IPsec just as you did in the earlier lesson "Implementing IPsec VPNs Using Pre-Shared Keys" on site-to-site VPNs using pre-shared keys.

5. Create Access Lists for Encryption Traffic (Crypto ACLs)
6. Create access lists for the traffic that you want to protect.
7. Configure Crypto Map
8. Apply Crypto Map to an Interface
9. Test and verify IPsec.

   Use **show**, **debug**, and related commands to test and verify that IPsec encryption works and to troubleshoot problems.

Preparing for IPsec

This section describes each of the steps used to prepare for IPsec configuration on a router.



Preparing for IPsec

```
R1# ping 172.30.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1# show crypto ipsec policy
R1# show crypto isakmp policy
Global IKE policy
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
R1# show crypto map
No crypto maps found.
R1# show crypto ipsec transform-set
R1#
```

SNRS v2.0—4-12

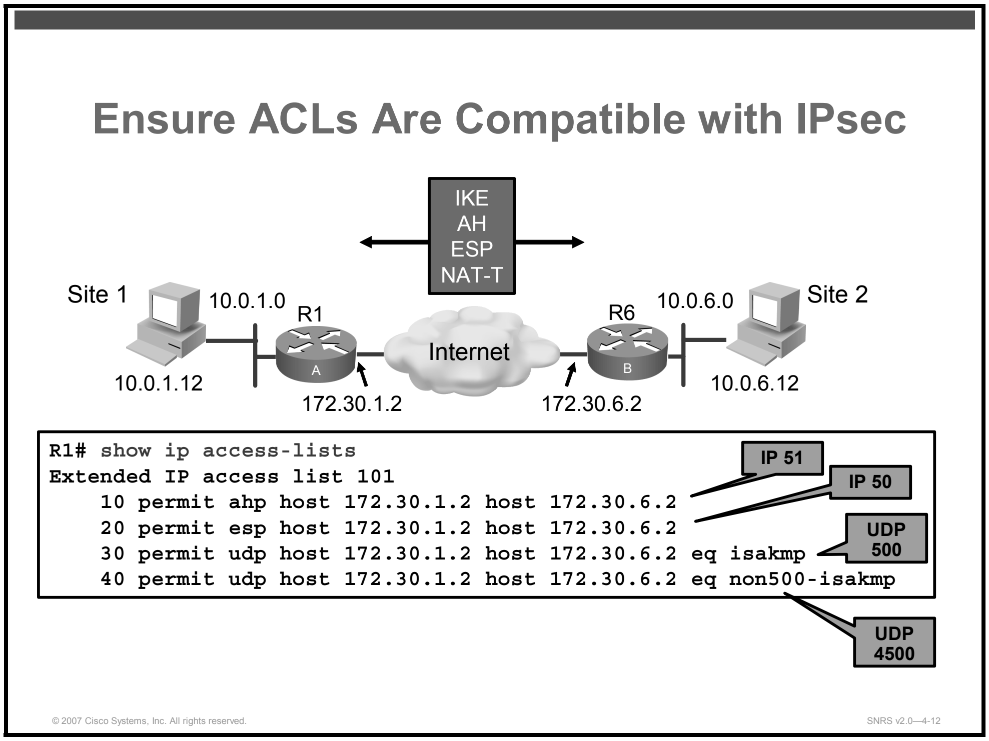
Preparing the network for IPsec is done exactly in the same way that you did with pre-shared keys.

Preparing for IPsec includes these tasks:

- Checking connectivity without IPsec configured

Ensure the network works without encryption.

- Checking for previous IPsec configurations
  - Checking for any existing ISAKMP or IPsec policies
  - Checking for any existing crypto maps or transform sets



Ensure ACLs Are Compatible with IPsec

```
R1# show ip access-lists
Extended IP access list 101
    10 permit ahp host 172.30.1.2 host 172.30.6.2
    20 permit esp host 172.30.1.2 host 172.30.6.2
    30 permit udp host 172.30.1.2 host 172.30.6.2 eq isakmp
    40 permit udp host 172.30.1.2 host 172.30.6.2 eq non500-isakmp
```

SNRS v2.0—4-12

ACLs must be compatible with IPsec. The ACLs must allow the following protocols through:

- ISAKMP

The ACL must allow UDP 500.

- Authentication Header

The ACL must allow IP 51.

- Encapsulation Security Payload

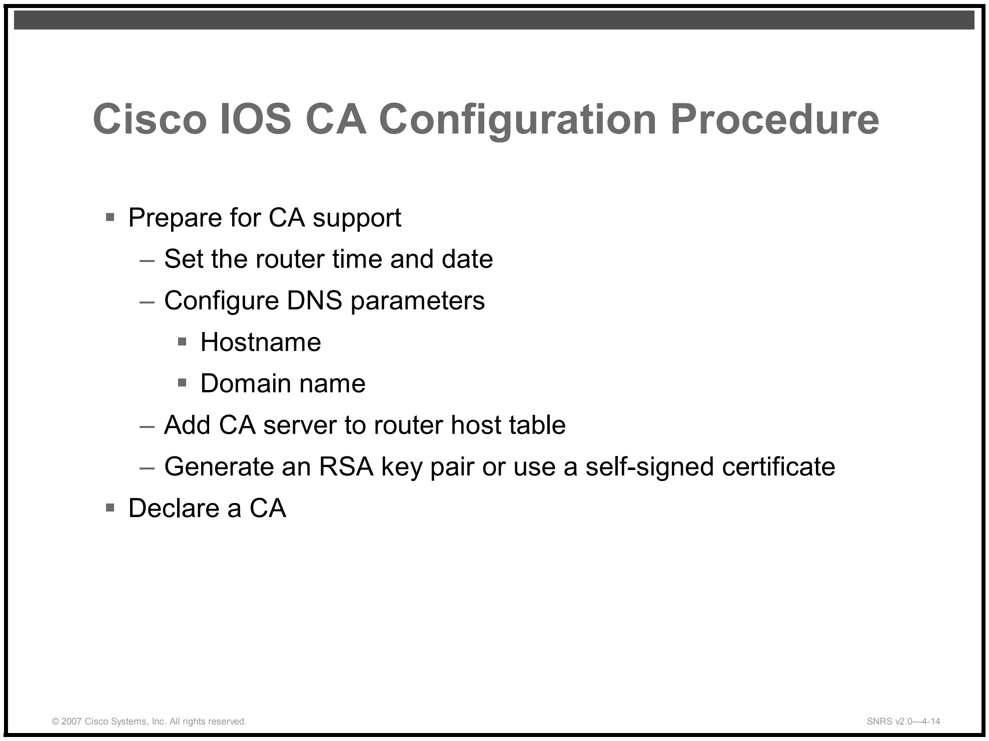
The ACL must allow IP 51.

- NAT-Traversal

The ACL must allow UDP 4500.

Configuring CA Support

This section describes the steps necessary to configure CA interoperability on a Cisco router.



Cisco IOS CA Configuration Procedure

- Prepare for CA support
  - – Set the router time and date
  - – Configure DNS parameters
    - Hostname
    - Domain name
  - – Add CA server to router host table
  - – Generate an RSA key pair or use a self-signed certificate
- Declare a CA

SNRS v2.0—4-14

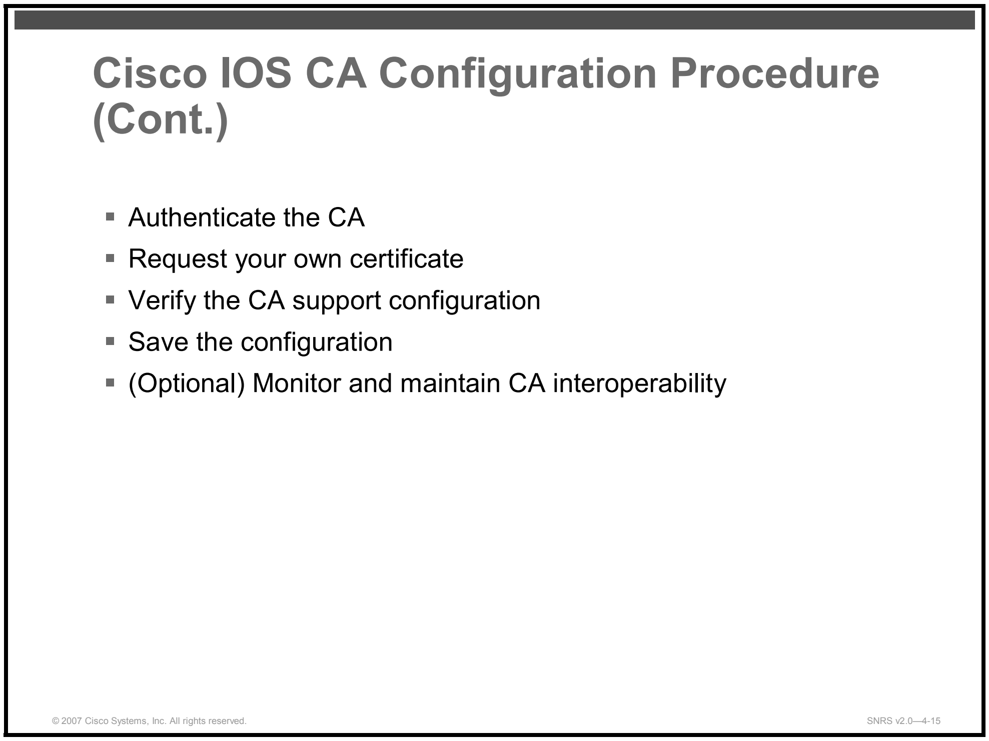
There are several steps required to configure a router to use PKI. Having a detailed plan lessens the chances of improper configuration. Some planning steps to prepare for CA support include the following:

Step 1 Set the router time and date.

Step 2 Configure the router hostname and domain name.

Step 3 Generate an RSA key pair. Digital signatures generated using RSA keys are used to authenticate the remote VPN peer. You can generate one general purpose key or two special purpose keys.

Step 4 Declare a CA. To declare the CA that your router should use, use the **crypto pki trustpoint** global configuration command. Use the **no** form of this command to delete all identity information and certificates associated with the CA.



Cisco IOS CA Configuration Procedure (Cont.)

- Authenticate the CA
- Request your own certificate
- Verify the CA support configuration
- Save the configuration
- (Optional) Monitor and maintain CA interoperability

SNRS v2.0—4-15

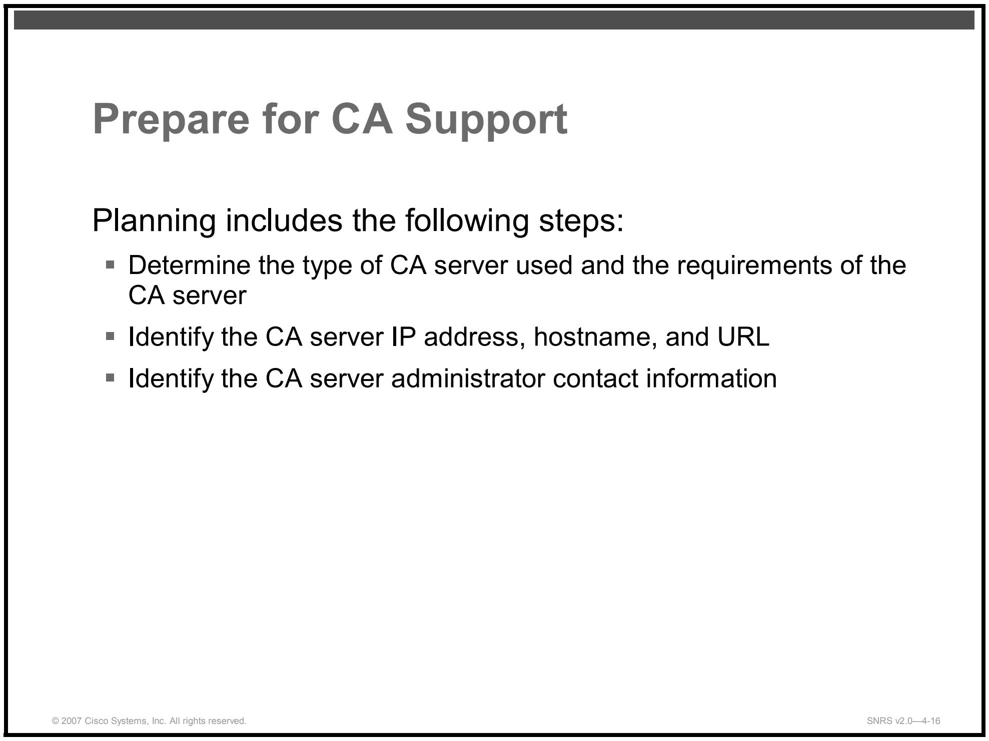
Step 5 Authenticate the CA. The router needs to authenticate the CA. It does this by obtaining the CA self-signed certificate that contains the CA public key.

Step 6 Request your own certificate. Complete this step to obtain the identity certificate for your router from the CA.

Step 7 Verify the CA support configuration. The commands detailed in this topic allow you to view your CA certificates and any other configured CA certificates.

Step 8 Save the configuration. After you have configured the router for CA support, the configuration should be saved.

Step 9 (Optional) Monitor and maintain CA interoperability.



Prepare for CA Support

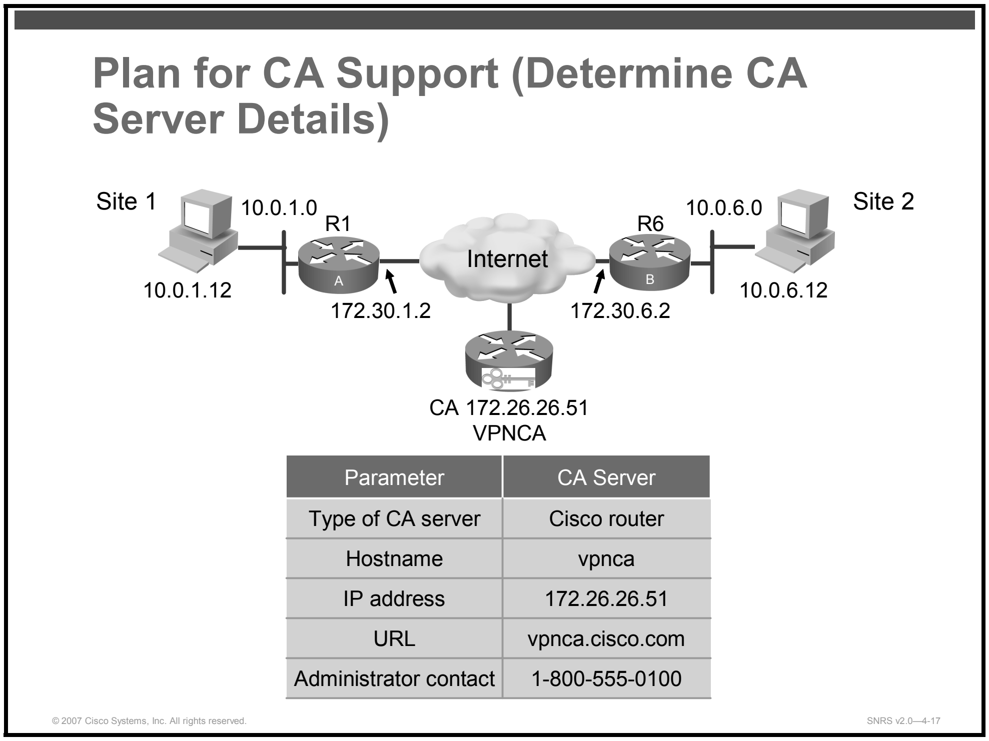
Planning includes the following steps:

- Determine the type of CA server used and the requirements of the CA server
- Identify the CA server IP address, hostname, and URL
- Identify the CA server administrator contact information

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-16

Configuring a CA is complicated. One of the first steps is to prepare to configure the CA parameters by gathering specific information that will be asked for during the configuration process. Having a detailed plan lessens the chances of improper configuration. Some planning steps include the following:

- Determine the type of CA server to use. CA servers come in a multitude of configurations and capabilities. You must determine which type of CA server fits your needs in advance of configuration. Requirements include (but are not limited to) the RSA key type required, CRL capabilities, and support for RA mode.
- Identify the CA server IP address, hostname, and URL. (This information is necessary if you use LDAP.)
- Identify the CA server administrator contact information. You need to arrange for your certificates to be validated if the process is not automatic.



Plan for CA Support (Determine CA Server Details)

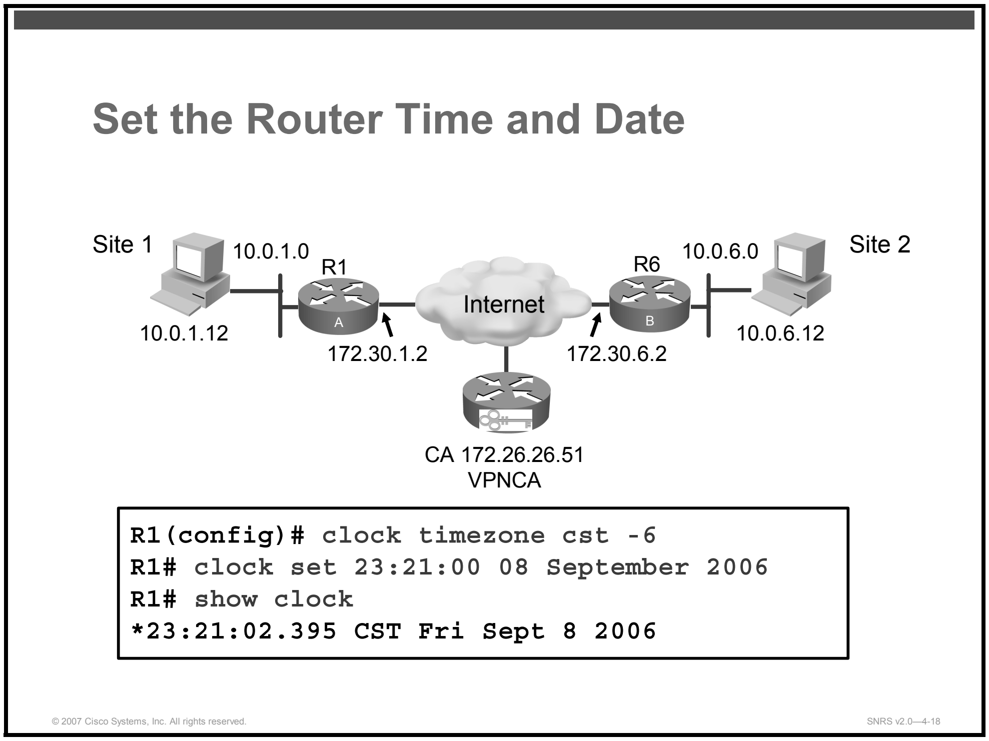
Parameter	CA Server
Type of CA server	Cisco router
Hostname	vpnca
IP address	172.26.26.51
URL	vpnca.cisco.com
Administrator contact	1-800-555-0100

SNRS v2.0—4-17

This figure illustrates the minimum information needed to configure a CA server on a Cisco router. Depending on the CA server chosen, other variables may also have to be identified and resolved.

Setting the Router Time and Date

This section describes the steps required to set the router clock.



Set the Router Time and Date

```
R1(config)# clock timezone cst -6
R1# clock set 23:21:00 08 September 2006
R1# show clock
*23:21:02.395 CST Fri Sept 8 2006
```

SNRS v2.0—4-18

The router must have a valid date and time configuration. Ensure that the time zone, time, and date for the router have been accurately set with the **show clock** command in privileged EXEC mode. The clock must be accurately set before generating RSA key pairs and enrolling with the CA server because certificates are time-sensitive. On certificates, there is a valid from and to date and time. When the certificate is validated by the router, the router determines if its system clock falls within the validity range. If it does, the certificate is valid. If not, the certificate is deemed invalid or expired.

Follow these steps to set the correct date and time on the router:

Step 1 Specify the router time zone.

router(config)# **clock timezone** *zone hours-offset* [*minutes-offset*]

Syntax Description

zone	Name of the time zone to be displayed when standard time is in effect The length of the *zone* argument is limited to 7 characters.
hours-offset	Hours difference from Coordinated Universal Time (UTC)
minutes-offset	(Optional) Minutes difference from UTC

The command sets the time zone and an offset from UTC (displayed by the router). The system internally keeps time in UTC, so this command is used only for display purposes and when the time is manually set.

Step 2 Set the router time and date.

```
router(config)# clock set hh:mm:ss day month year
```

Syntax Description

hh:mm:ss	Current time in hours (24-hour format), minutes, and seconds
day	Current day (by date) in the month
month	Current month (by name)
year	Current year (no abbreviation)

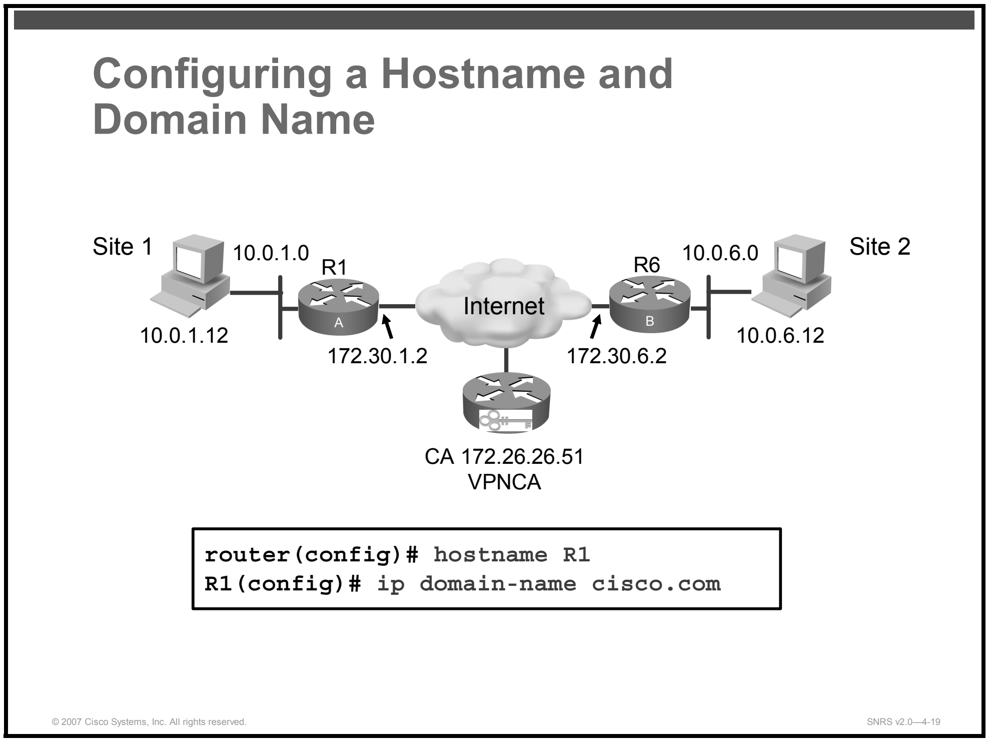
Generally, if the system is synchronized by a valid outside timing mechanism, such as an NTP or Banyan Virtual Integrated Network Service (VINES) clock source, or if you have a router with a hardware clock, you need not set the software clock. Use this command if no other time sources are available. The time specified in this command is assumed to be in the time zone specified by the configuration of the **clock timezone** command.

You can also optionally set your router to automatically update the calendar and time from an NTP server with the **ntp** series of commands.

Note It is recommended that you use an NTP server to synchronize the router clock.

Configuring DNS Parameters

This section describes how to give the router a host name and domain name and how to add a CA server entry to the router host table.



Configuring a Hostname and Domain Name

```
router(config)# hostname R1
R1(config)# ip domain-name cisco.com
```

SNRS v2.0—4-19

You must configure the hostname and IP domain name of the router if this has not already been done. This process is required because the router assigns a fully qualified domain name (FQDN) to the keys and certificates used by IPsec, and the FQDN is based on the hostname and IP domain name that you assign to the router. For example, a certificate named “router20.example.com” is based on a router hostname of “router20” and a router IP domain name of “example.com.”

Follow these steps to configure Domain Name System (DNS) parameters used with PKI:

Step 1 Specify or modify the hostname for the network server. The hostname is used in prompts and default configuration filenames. The setup command facility also prompts for a hostname at startup.

```
router(config)# hostname name
```

Syntax Description

name	New hostname for the router

The hostname is used in prompts and default configuration filenames.

Do not expect case to be preserved. Uppercase and lowercase characters look the same to many Internet software applications. It may seem appropriate to capitalize a name the same way that you might in English, but conventions dictate that computer names appear all lowercase. For more information, refer to RFC 1178, *Choosing a Name for Your Computer*.

The name must also follow the rules for Advanced Research Projects Agency Network (ARPANET) hostnames. Hostnames must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphens. Hostnames must be 63 characters or fewer. A hostname of less than 10 characters is recommended. For more information, refer to RFC 1035, *Domain Names—Implementation and Specification*.

On most systems, a field of 30 characters is used for the hostname and the prompt in the command-line interface (CLI). Note that the length of your hostname may cause longer configuration mode prompts to be truncated.

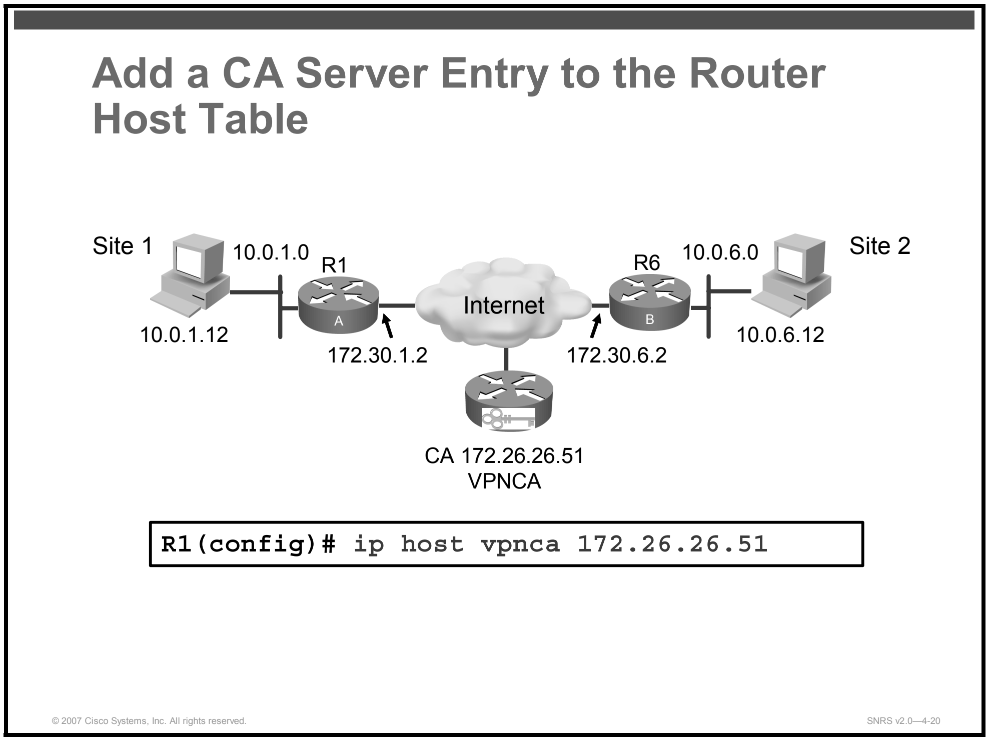
Step 2 Define a default domain name that the Cisco IOS Software uses to complete unqualified hostnames (names without a dotted decimal domain name).

```
router(config)# ip domain-name name
```

Syntax Description

name	Default domain name used to complete unqualified hostnames Do not include the initial period that separates an unqualified name from the domain name.

Any IP hostname that does not contain a domain name (that is, any name without a dot), will have the dot and cisco.com appended to it before being added to the host table.



Add a CA Server Entry to the Router Host Table

```
R1(config)# ip host vpnca 172.26.26.51
```

SNRS v2.0—4-20

You can map IP addresses to hostnames for DNS purposes. Use the **ip host** global configuration command to define a static hostname-to-address mapping in the host cache. To remove the name-to-address mapping, use the **no** form of this command.

Step 3 Define a static host to IP address mapping for the CA server.

```
router(config)# ip host [vrf vrf-name] {name | tmodem-
telephone-number} [tcp-port-number] address1
[address2...address8]
```

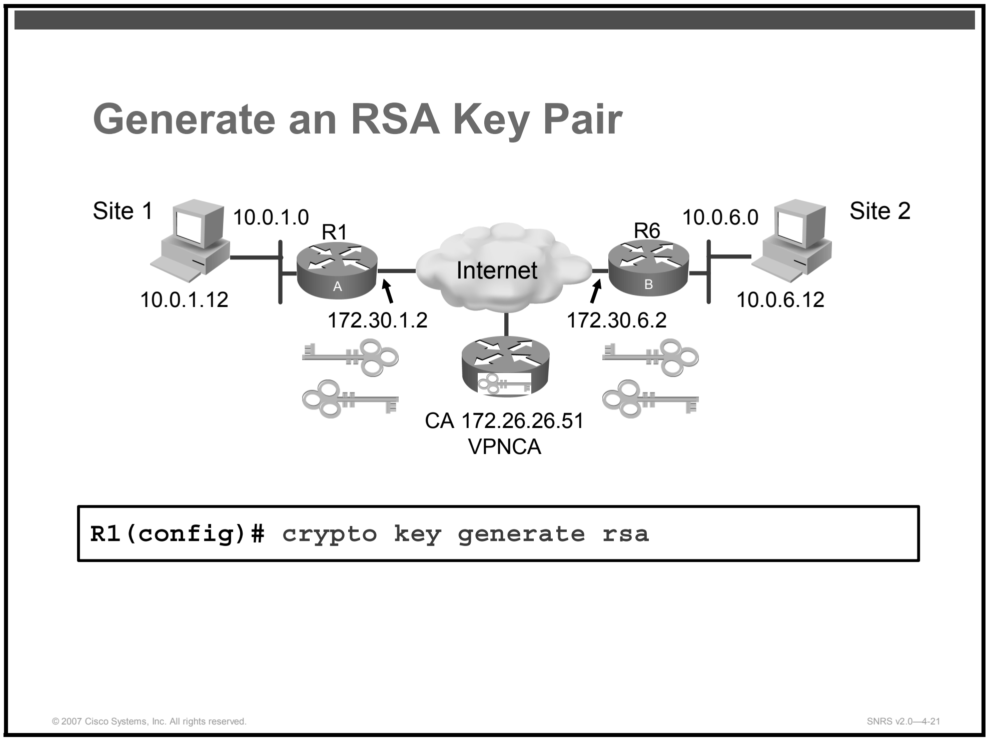
Syntax Description

vrf *vrf-name*	(Optional) Defines a VPN routing and forwarding VRF table. The *vrf-name* argument specifies a name for the VRF table.
name	Name of the host. The first character can be either a letter or a number. If you use a number, the types of operations that you can perform are limited.
t*modem-telephone-number*	Modem telephone number that is mapped to the IP host address for use in Cisco modem user interface mode. You must enter the letter "t" before the telephone number.
tcp-port-number	(Optional) TCP port number to connect to when using the defined hostname in conjunction with an EXEC connect or Telnet command. The default is Telnet (port 23).
address1	Associated IP host address.
address2...address8	(Optional) Additional associated IP addresses. You can bind up to eight addresses to a hostname.

The first character can be either a letter or a number. If you use a number, the types of operations that you can perform (such as **ping**) are limited.

Generating an RSA Key Pair

This topic describes how to generate an RSA key pair to be used with PKI.



Generate an RSA Key Pair

```
R1(config)# crypto key generate rsa
```

SNRS v2.0—4-21

RSA key pairs are used to sign and encrypt (IKE key management messages and are required before you can obtain a certificate for your router.

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

If you want a modulus value between 512 and 1024, enter an integer value that is a multiple of 64. If you want a value higher than 1024, enter 1536 or 2048. If you enter a value greater than 512, key generation may take a minute or longer.

Usage RSA Keys Vs. General Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs. When you generate RSA key pairs (via the **crypto key generate rsa** command), you will be prompted to choose either usage keys or general purpose keys.

If you generate special usage keys, two pairs of RSA keys will be generated. One pair will be used with any IKE policy that specifies RSA signatures as the authentication method, and the other pair will be used with any IKE policy that specifies RSA encrypted keys as the authentication method.

A CA is used only with IKE policies specifying RSA signatures, not with IKE policies specifying RSA-encrypted nonces. (However, you could specify more than one IKE policy and have RSA signatures specified in one policy and RSA-encrypted nonces in another policy.)

If you plan to have both types of RSA authentication methods in your IKE policies, you may prefer to generate special usage keys. With special usage keys, each key is not unnecessarily exposed.

Note Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.

If you generate general purpose keys, only one pair of RSA keys will be generated. This pair will be used with IKE policies specifying either RSA signatures or RSA-encrypted keys. Therefore, a general purpose key pair might get used more frequently than a special usage key pair.

Named Key Pairs

If you generate a named key pair using the *key-pair-label* argument, you must also specify the **usage-keys** keyword or the **general-keys** keyword. Named key pairs allow you to have multiple RSA key pairs, enabling the Cisco IOS Software to maintain a different key pair for each identity certificate.

Modulus Length

When you generate RSA keys, you will be prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate and takes longer to use.

Note The Cisco IOS Software does not support a modulus greater than 2048 bits.

A length of less than 512 is normally not recommended.

Note In certain situations, the shorter modulus may not function properly with IKE, so Cisco recommends using a minimum modulus of 1024.

Generating RSA Keys

Use the **crypto key generate rsa** global configuration command to generate RSA key pairs.

```
router(config)# crypto key generate rsa {general-keys | usage-
keys} [label key-label] [exportable] [modulus modulus-size]
[storage device:]
```

Syntax Description

general-keys	Specifies that the general purpose key pair should be generated
usage-keys	Specifies that two RSA special usage key pairs should be generated (that is, one encryption pair and one signature pair) instead of one general purpose key pair
label *key-label*	(Optional) Name that is used for an RSA key pair when the key pair is being exported If a key label is not specified, the FQDN of the router is used.
exportable	(Optional) Specifies that the RSA key pair can be exported to another Cisco device, such as a router
modulus *modulus-size*	(Optional) IP size of the key modulus in a range from 350 to 2048 If you do not enter the modulus keyword and specify a size, you will be prompted.
storage *device:*	(Optional) Specifies the key storage location The name of the storage device is followed by a colon (:).

Use this command to generate RSA key pairs for your Cisco device (such as a router).

RSA keys are generated in pairs—one public RSA key and one private RSA key.

If your router already has RSA keys when you issue this command, you will be warned and prompted to replace the existing keys with new keys.

Note Before issuing this command, ensure that your router has a hostname and IP domain name configured (with the **hostname** and **ip domain-name** commands). You will be unable to complete the **crypto key generate rsa** command without a hostname and IP domain name. (This situation is not true when you only generate a named key pair.)

Note SSH may generate an additional RSA key pair if you generate a key pair on a router having no RSA keys. The additional key pair is used only by SSH and will have a name such as “{router_FQDN}.server”. For example, if a router’s fully qualified domain name (FQDN) is "router1.cisco.com," the key name is "router1.cisco.com.server."

This command is not saved in the router configuration; however, the RSA keys generated by this command are saved in the private configuration in NVRAM (which is never displayed to the user or backed up to another device).

There are two mutually exclusive types of RSA key pairs: special-usage keys and general-purpose keys. When you generate RSA key pairs, you will be prompted to select either special-usage keys or general-purpose keys.

Removing RSA Key Pairs

You might want to remove an RSA key pair for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so you would have to delete the old 1024-bit keys and generate new 2048-bit keys.

To remove all RSA keys or the specified RSA key pair that has been generated by your router, use the following command:

```
router(config)# crypto key zeroize rsa [key-pair-label]
```

Syntax Description

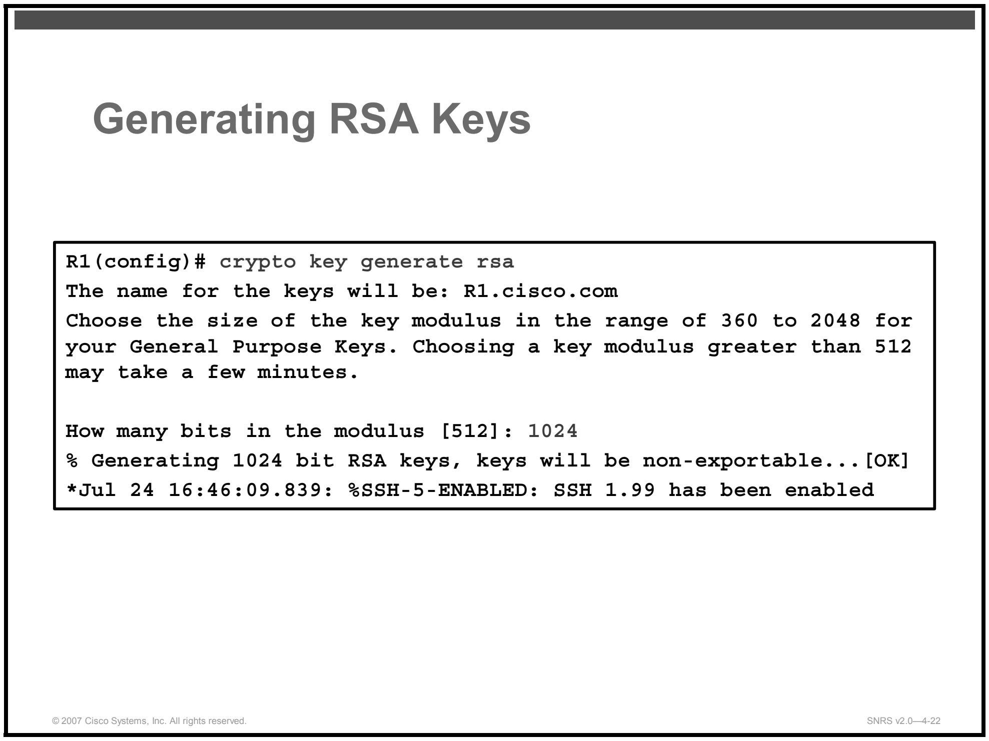
key-pair-label	(Optional) Specifies the name of the key pair that the router will delete

Note If the *key-pair-label* argument is not specified, all RSA keys that have been generated by your router will be deleted.

This command deletes all RSA keys that were previously generated by your router unless you include the *key-pair-label* argument, which will delete only the specified RSA key pair. If you issue this command, you must also perform two additional tasks for each trustpoint that is associated with the key pair that was deleted:

- Ask the CA administrator to revoke your router certificates at the CA; you must supply the challenge password that you created when you originally obtained the router certificates using the **crypto ca enroll** command.
- Manually remove the router certificates from the configuration by removing the configured trustpoint (using the **no crypto ca trustpoint** *name* command.)

Note This command cannot be undone (after you save your configuration). After RSA keys have been deleted, you cannot use certificates or the CA or participate in certificate exchanges with other IPsec peers unless you reconfigure CA interoperability by regenerating RSA keys, getting the CA certificate, and requesting your own certificate again.



Generating RSA Keys

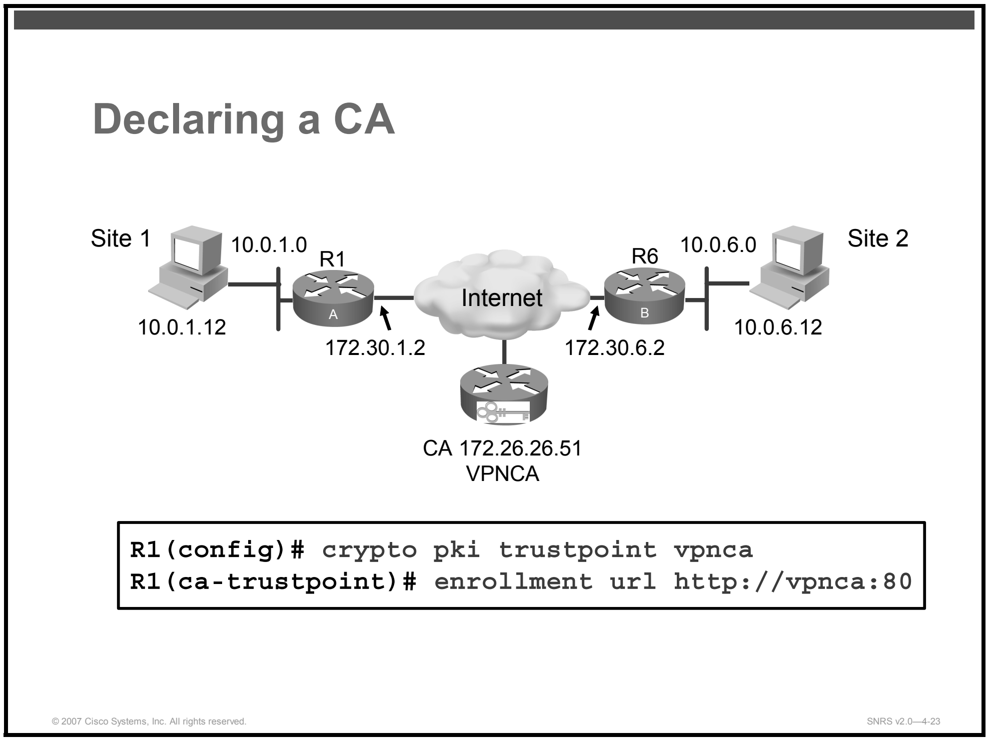
```
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for
your General Purpose Keys. Choosing a key modulus greater than 512
may take a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Jul 24 16:46:09.839: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

SNRS v2.0—4-22

This figure shows an example of RSA key generation. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus could offer stronger security but takes longer to generate and takes longer to use. A modulus below 512 is normally not recommended. It is recommended that you use a minimum modulus of 1024.

Declaring a CA

This topic describes the process of declaring a CA.



Declaring a CA

```
R1(config)# crypto pki trustpoint vpnca
R1(ca-trustpoint)# enrollment url http://vpnca:80
```

SNRS v2.0—4-23

The example shown in the figure declares a CA and identifies characteristics of the CA. In this example, the name "vpnca" is created for the CA, which is located at http://vpnca port 80. This is the minimum possible configuration required to declare a CA.

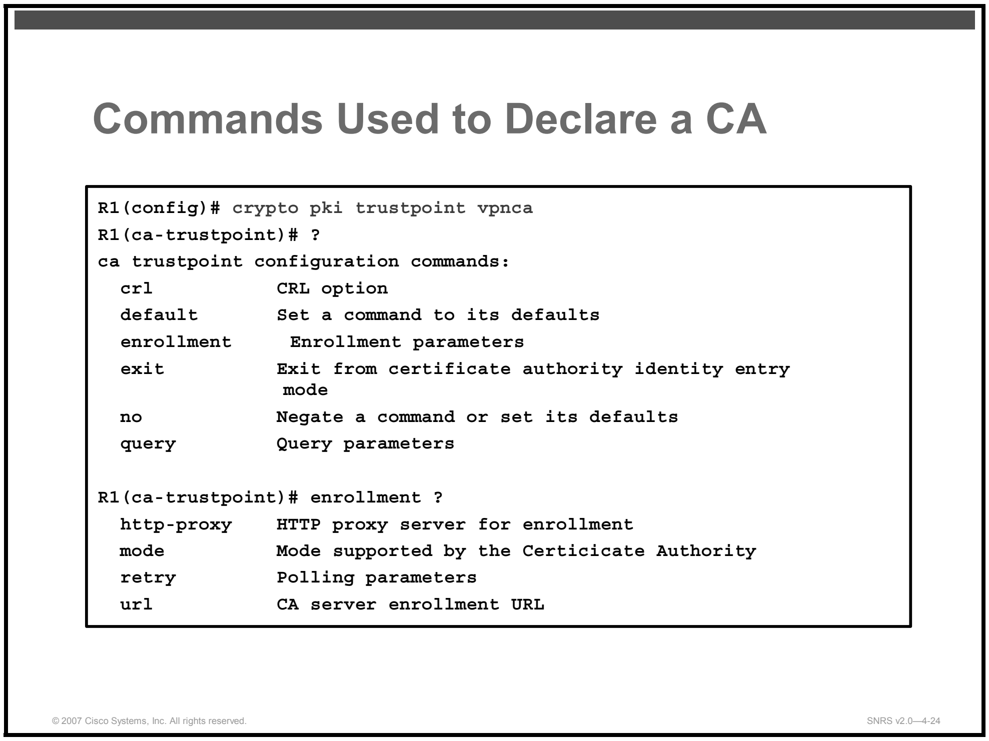
Follow these steps to declare a CA server:

Step 1 Declare which CA your router will use. Issuing the **crypto pki trustpoint** command puts you in CA trustpoint configuration mode.

```
router(config)# crypto pki trustpoint name
```

Syntax Description

name	Creates a name for the trustpoint (If you previously declared the trustpoint and just want to update its characteristics, specify the name that you previously created.)



Commands Used to Declare a CA

```
R1(config)# crypto pki trustpoint vpnca
R1(ca-trustpoint)# ?
ca trustpoint configuration commands:
  crl           CRL option
  default       Set a command to its defaults
  enrollment     Enrollment parameters
  exit          Exit from certificate authority identity entry
                 mode
  no            Negate a command or set its defaults
  query         Query parameters

R1(ca-trustpoint)# enrollment ?
  http-proxy    HTTP proxy server for enrollment
  mode          Mode supported by the Certicicate Authority
  retry         Polling parameters
  url           CA server enrollment URL
```

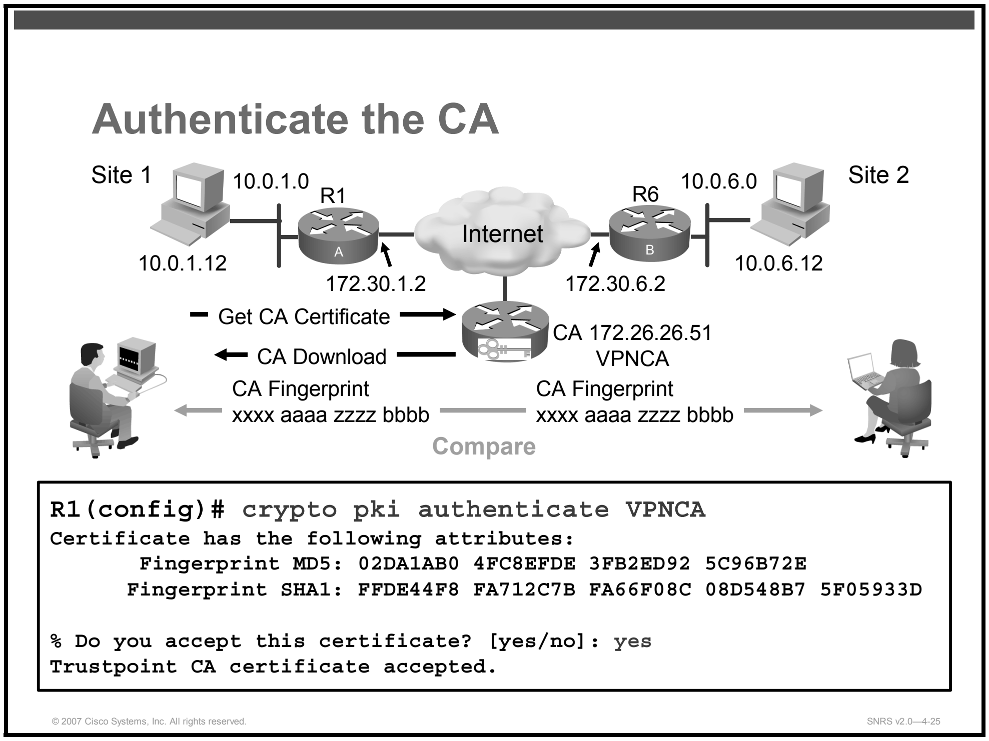
SNRS v2.0—4-24

Performing the **crypto pki trustpoint** command puts you into the CA trustpoint configuration mode, where you can specify characteristics for the CA with the following commands:

- **crl:** Queries the CRL to ensure that the certificate of the peer has not been revoked
- **default (ca-trustpoint):** Resets the value of CA trustpoint configuration mode subcommands to their defaults
- **query url:** Specify the URL of the LDAP server (required only if your CA supports an RA and the LDAP protocol)
- **enrollment:** Specifies enrollment parameters (optional)
- **enrollment http-proxy:** Accesses the CA by HTTP through the proxy server
- **enrollment selfsigned:** Specifies self-signed enrollment (optional)
- **enrollment mode:** Specifies the RA mode (required only if your CA system provides an RA)
- **enrollment url:** Specifies the URL of the CA (always required)
- **match certificate**: Associates a certificate-based ACL defined with the **crypto ca certificate map** command
- **ocsp disable-nonce**: Specifies that your router will not send unique identifiers, or nonces, during OCSP communications

Authenticating the CA

This section describes how to authenticate the CA.



Authenticate the CA

```
R1(config)# crypto pki authenticate VPNCA
Certificate has the following attributes:
       Fingerprint MD5: 02DA1AB0 4FC8EFDE 3FB2ED92 5C96B72E
      Fingerprint SHA1: FFDE44F8 FA712C7B FA66F08C 08D548B7 5F05933D

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
```

The router needs to authenticate the CA to verify that it is valid. The router does this by obtaining the CA self-signed certificate that contains the public key of the CA. Because the CA certificate is self-signed (the CA signs its own certificate), the CA certificate should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate when you perform this step. To get the public key of the CA, use the **crypto pki authenticate** *name* command in global configuration mode. Use the same name that you used when declaring the CA with the **crypto pki trustpoint** command.

The following example shows a CA authentication:

```
router(config)# crypto pki authenticate name
```

Syntax Description

name	Specifies the name of the CA This is the same name used when the CA was declared with the **crypto ca identity** command.

This command is required when you initially configure CA support at your router.

This command authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA. Because the CA signs its own certificate, you should manually authenticate the public key of the CA by contacting the CA administrator when you perform this command.

If you are using RA mode (using the **enrollment mode ra** command) when you issue the **crypto pki authenticate** command, RA signing and encryption certificates will be returned from the CA and the CA certificate.

This command is not saved to the router configuration. However, the public keys embedded in the received CA (and RA) certificates are saved to the configuration as part of the RSA public key record (called the RSA public key chain).

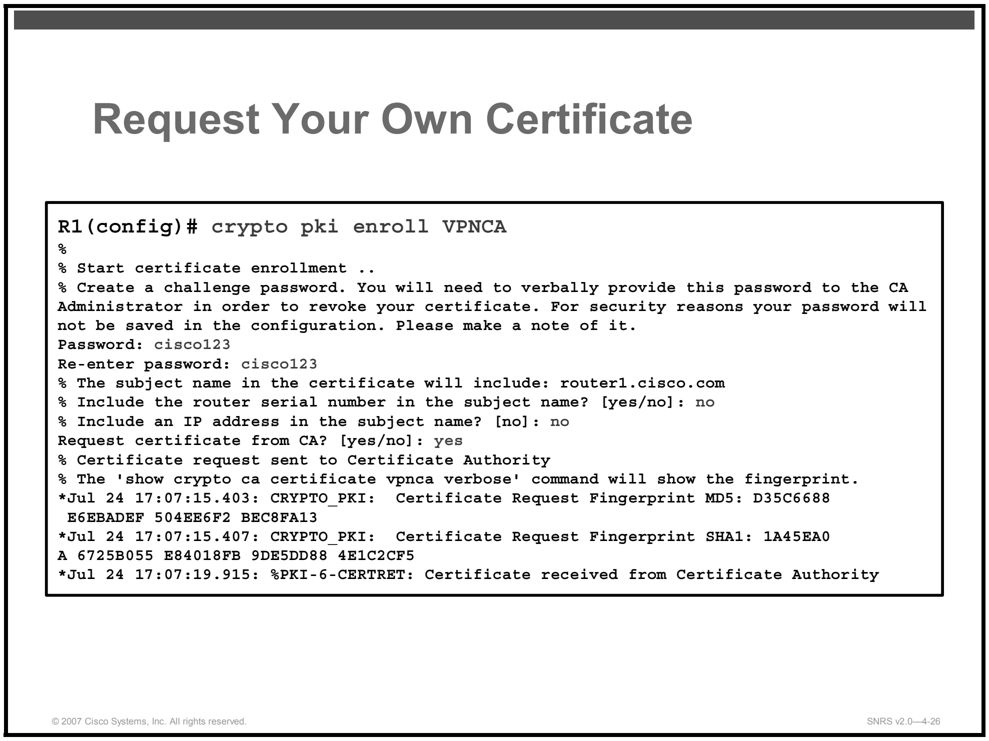
If the CA does not respond by a timeout period after this command is issued, the terminal control will be returned so that it will not be tied up. If this happens, you must re-enter the command. Cisco IOS Software will not recognize CA certificate expiration dates set for beyond the year 2049. If the validity period of the CA certificate is set to expire after the year 2049, the following error message will be displayed when authentication with the CA server is attempted:

```
error retrieving certificate :incomplete chain
```

If you receive an error message similar to this one, check the expiration date of your CA certificate. If the expiration date of your CA certificate is set after the year 2049, you must reduce the expiration date by a year or more.

Requesting a Certificate from a CA

This section describes the process for requesting a certificate from a CA.



Request Your Own Certificate

```
R1(config)# crypto pki enroll VPNCA
%
% 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: cisco123
Re-enter password: cisco123
% The subject name in the certificate will include: router1.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate vpnca verbose' command will show the fingerprint.
*Jul 24 17:07:15.403: CRYPTO_PKI:  Certificate Request Fingerprint MD5: D35C6688
 E6EBADEF 504EE6F2 BEC8FA13
*Jul 24 17:07:15.407: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 1A45EA0
A 6725B055 E84018FB 9DE5DD88 4E1C2CF5
*Jul 24 17:07:19.915: %PKI-6-CERTRET: Certificate received from Certificate Authority
```

SNRS v2.0—4-26

You must obtain a signed certificate from the CA for each of the RSA key pairs of your router. If you generated general purpose RSA keys, your router has only one RSA key pair and needs only one certificate. If you previously generated special usage RSA keys, your router has two RSA key pairs and needs two certificates.

To obtain the certificate (or certificates) for your router from the CA, use the **crypto pki enroll** command in global configuration mode. To delete a current enrollment request, use the **no** form of this command.

```
router(config)# crypto pki enroll name
```

Syntax Description

name	Specifies the name of the CA Use the same name as when you declared the CA using the **crypto pki trustpoint** command.

This command requests certificates from the CA for all of your router's Rivest, Shamir, and Adelmen (RSA) key pairs. This task is also known as enrolling with the CA. (Technically, enrolling and obtaining certificates are two separate events, but they both occur when this command is issued.)

Your router needs a signed certificate from the CA for each RSA key pair of your router; if you previously generated general purpose keys, this command obtains the one certificate corresponding to the one general purpose RSA key pair. If you previously generated special usage keys, this command obtains two certificates corresponding to each of the special usage RSA key pairs.

If you already have a certificate for your keys, you are unable to complete this command; instead, you are prompted to remove the existing certificate first. (You can remove existing certificates with the **no certificate** command.)

The **crypto pki enroll** command is not saved in the router configuration.

Enrollment Process

The **crypto pki enroll** command requests certificates from the CA for all of the RSA key pairs of your router. This task is also known as enrolling with the CA. (Technically, enrolling and obtaining certificates are two separate events, but they both occur when this command is issued.)

If you previously generated general purpose keys, this command obtains the one certificate corresponding to the one general purpose RSA key pair. If you previously generated special usage keys, this command obtains two certificates corresponding to each of the special usage RSA key pairs.

If you already have a certificate for your keys, you are unable to complete this command; instead, you are prompted to remove the existing certificate first.

Note You can remove existing certificates with the **no certificate** command.

Responding to Prompts

When you issue the **crypto pki enroll** command, you are prompted a number of times.

First, you are prompted to create a challenge password. This password can be up to 80 characters in length. This password is necessary in the event that you ever need to revoke your router certificate (or certificates). When you ask the CA administrator to revoke your certificate, you must supply this challenge password as a protection against fraudulent or mistaken revocation requests. You will need to manually provide this password to the CA administrator to revoke your certificate

Caution This password is not stored anywhere; therefore, you need to remember this password.

If you lose the password, the CA administrator may still be able to revoke the router certificate but will require further manual authentication of the router administrator identity.

You are also prompted to indicate whether the serial number of your router should be included in the obtained certificate. The serial number is not used by IPsec or IKE, but may be used by the CA to either authenticate certificates or to later associate a certificate with a particular router. Ask your CA administrator if serial numbers should be included. If you are in doubt, include the serial number.

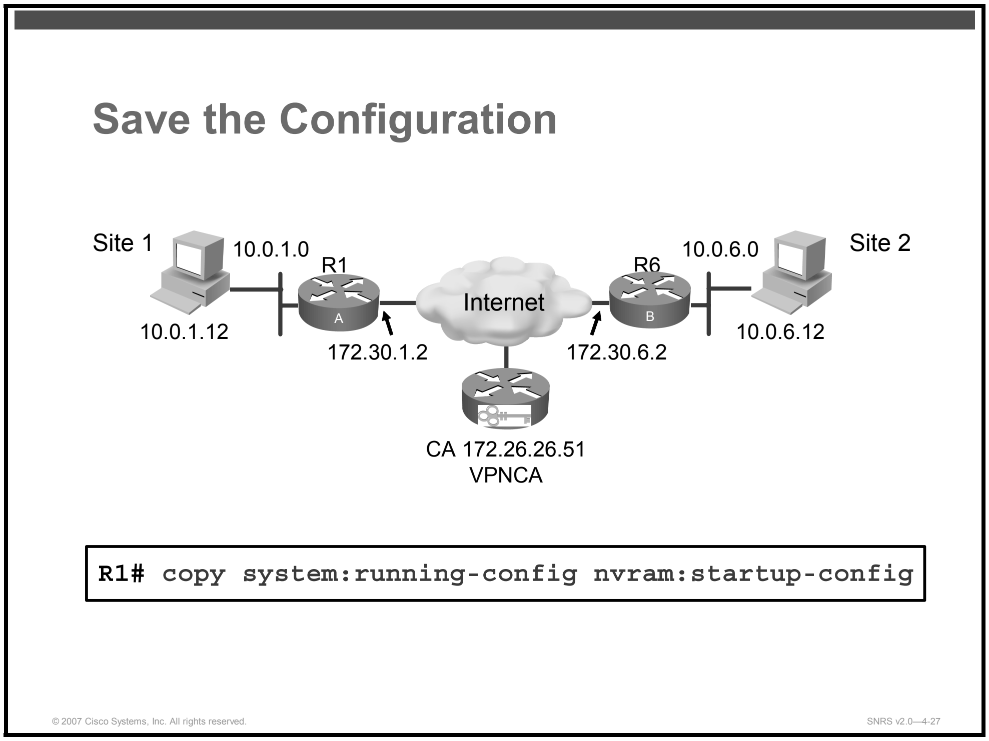
Normally, you would not include the IP address because the IP address binds the certificate more tightly to a specific entity. Also, if the router is moved, you would need to issue a new certificate. Finally, a router has multiple IP addresses, any of which might be used with IPsec.

If you indicate that the IP address should be included, you will then be prompted to specify the interface of the IP address. This interface should correspond to the interface to which you apply your crypto map set. If you apply crypto map sets to more than one interface, specify the interface that you name in the **crypto map local-address** command.

The figure shows an example enrollment session.

Saving the Configuration

This section describes how to save the running configuration to NVRAM.



Save the Configuration

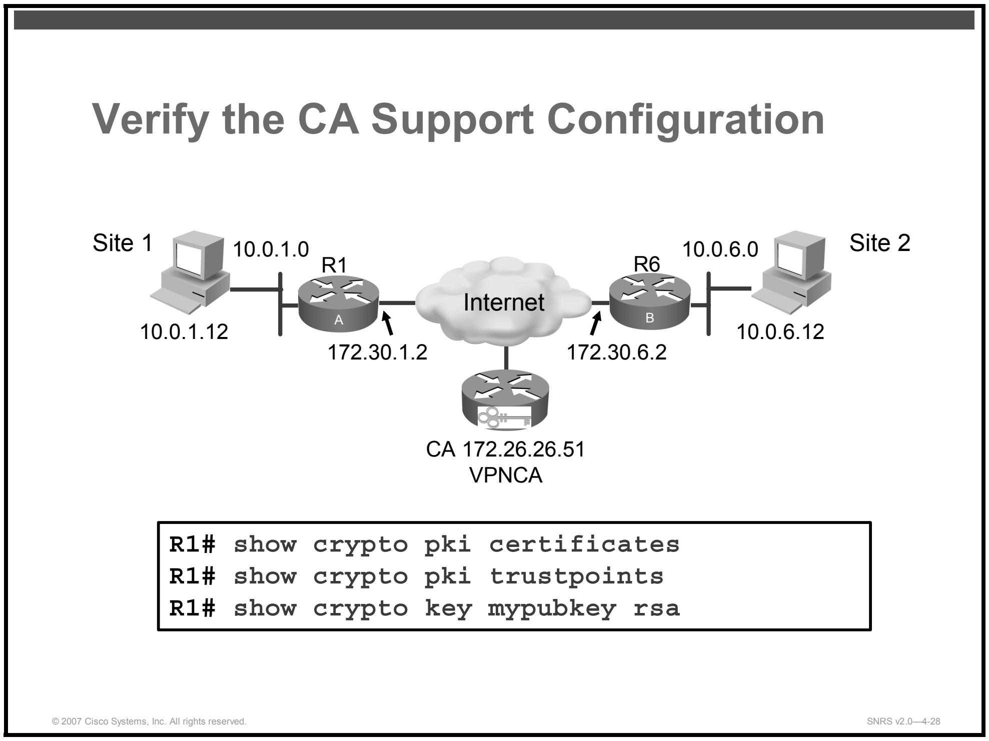
```
R1# copy system:running-config nvram:startup-config
```

SNRS v2.0—4-27

You must save the running configuration to NVRAM if you want to save the RSA key pairs and certificates on the local router. Use the **copy system:running-config nvram:startup-config** command to save your configuration. This command includes saving RSA keys to private NVRAM. RSA keys are *not* saved with your configuration when you use a **copy system:running-config rcp:** command or **copy system:running-config tftp:** command.

Verifying CA Support Configuration

This section describes how to verify a router configuration for CA support.

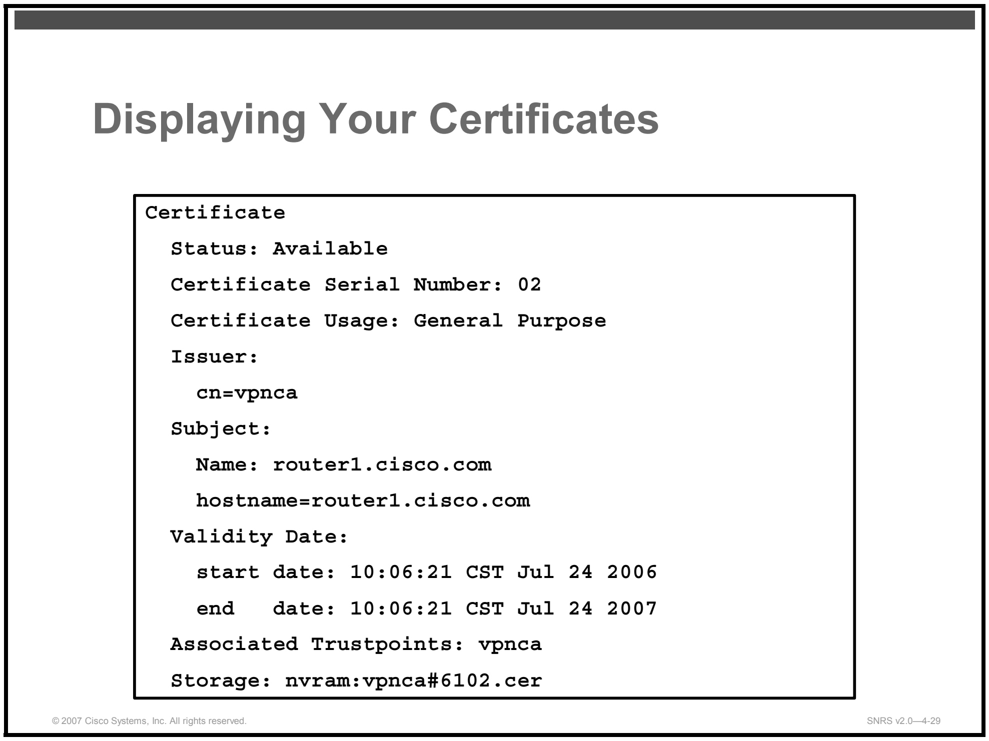


Verify the CA Support Configuration

```
R1# show crypto pki certificates
R1# show crypto pki trustpoints
R1# show crypto key mypubkey rsa
```

Here are some commands used to verify your CA support configuration:

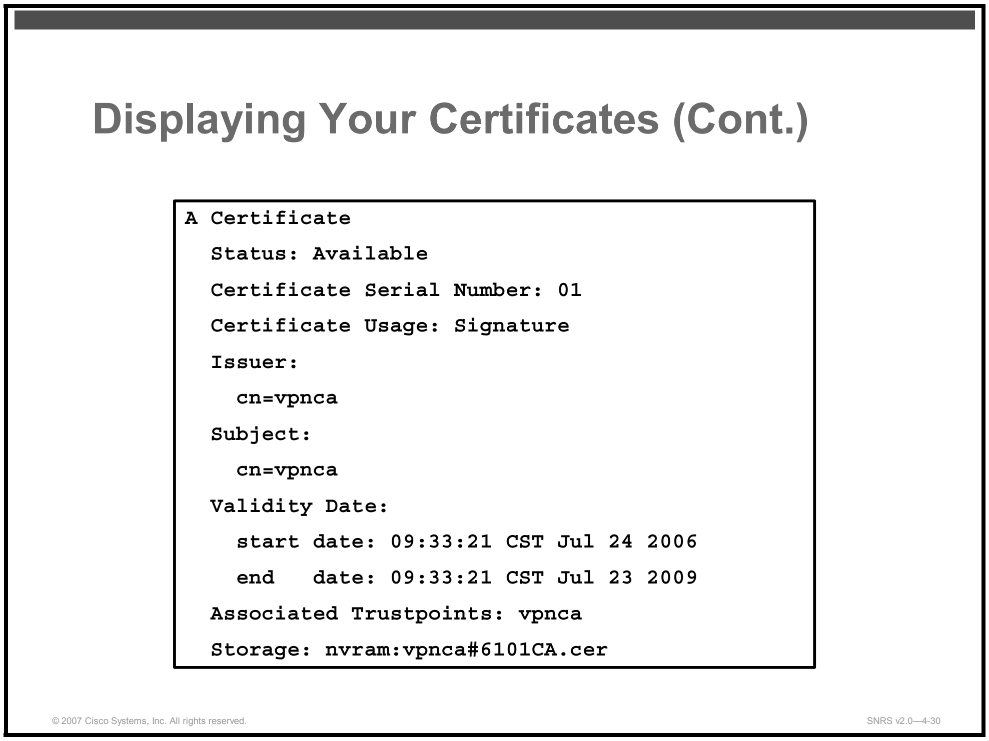
- **show crypto pki certificates:** Displays information about your certificate, the CA certificate, and any RA certificates
- **show crypto pki trustpoints:** Displays the trustpoints that are configured in the router
- **show crypto key mypubkey rsa:** Displays the RSA public keys of your router
- **show crypto key pubkey-chain rsa:** Displays a list of all the RSA public keys stored on your router (These include the public keys of peers that have sent your router their certificates during peer authentication for IPsec.)



Displaying Your Certificates

```
Certificate
  Status: Available
  Certificate Serial Number: 02
  Certificate Usage: General Purpose
  Issuer:
    cn=vpnca
  Subject:
    Name: router1.cisco.com
    hostname=router1.cisco.com
  Validity Date:
    start date: 10:06:21 CST Jul 24 2006
    end   date: 10:06:21 CST Jul 24 2007
  Associated Trustpoints: vpnca
  Storage: nvram:vpnca#6102.cer
```

SNRS v2.0—4-29

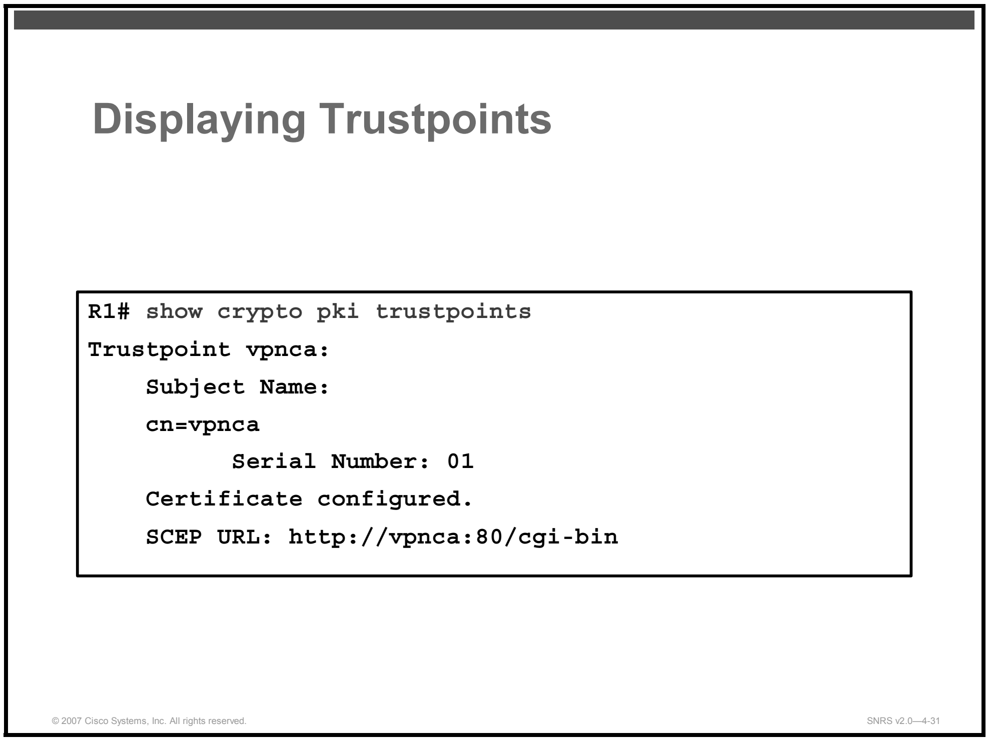


Displaying Your Certificates (Cont.)

```
A Certificate
  Status: Available
  Certificate Serial Number: 01
  Certificate Usage: Signature
  Issuer:
    cn=vpnca
  Subject:
    cn=vpnca
  Validity Date:
    start date: 09:33:21 CST Jul 24 2006
    end   date: 09:33:21 CST Jul 23 2009
  Associated Trustpoints: vpnca
  Storage: nvram:vpnca#6101CA.cer
```

SNRS v2.0—4-30

These two figures show an example of the **show crypto pki certificates** command. This command displays any certificates stored on the router. Notice that the certificates have been saved to NVRAM.

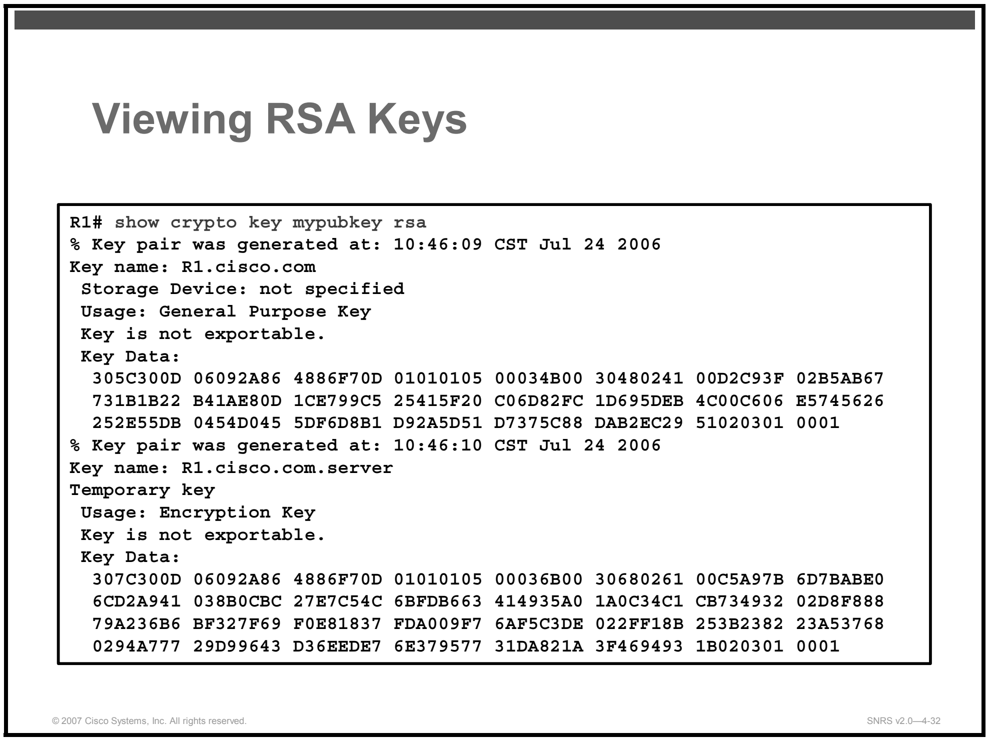


Displaying Trustpoints

```
R1# show crypto pki trustpoints
Trustpoint vpnca:
    Subject Name:
    cn=vpnca
          Serial Number: 01
    Certificate configured.
    SCEP URL: http://vpnca:80/cgi-bin
```

SNRS v2.0—4-31

This figure shows an example of the **show crypto pki trustpoints** command.



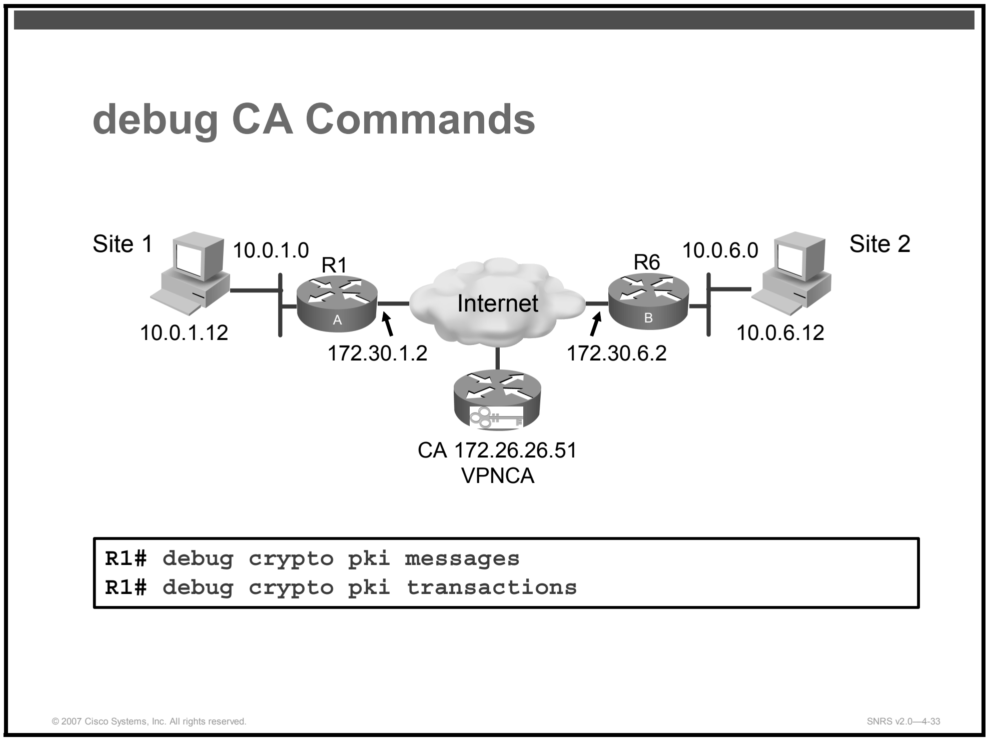
Viewing RSA Keys

```
R1# show crypto key mypubkey rsa
% Key pair was generated at: 10:46:09 CST Jul 24 2006
Key name: R1.cisco.com
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D2C93F 02B5AB67
  731B1B22 B41AE80D 1CE799C5 25415F20 C06D82FC 1D695DEB 4C00C606 E5745626
  252E55DB 0454D045 5DF6D8B1 D92A5D51 D7375C88 DAB2EC29 51020301 0001
% Key pair was generated at: 10:46:10 CST Jul 24 2006
Key name: R1.cisco.com.server
Temporary key
 Usage: Encryption Key
 Key is not exportable.
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C5A97B 6D7BABE0
  6CD2A941 038B0CBC 27E7C54C 6BFDB663 414935A0 1A0C34C1 CB734932 02D8F888
  79A236B6 BF327F69 F0E81837 FDA009F7 6AF5C3DE 022FF18B 253B2382 23A53768
  0294A777 29D99643 D36EEDE7 6E379577 31DA821A 3F469493 1B020301 0001
```

This figure shows an example of the **show crypto key mypubkey rsa** command.

Troubleshooting CA Interoperability

This section describes some commands used to troubleshoot CA interoperability.



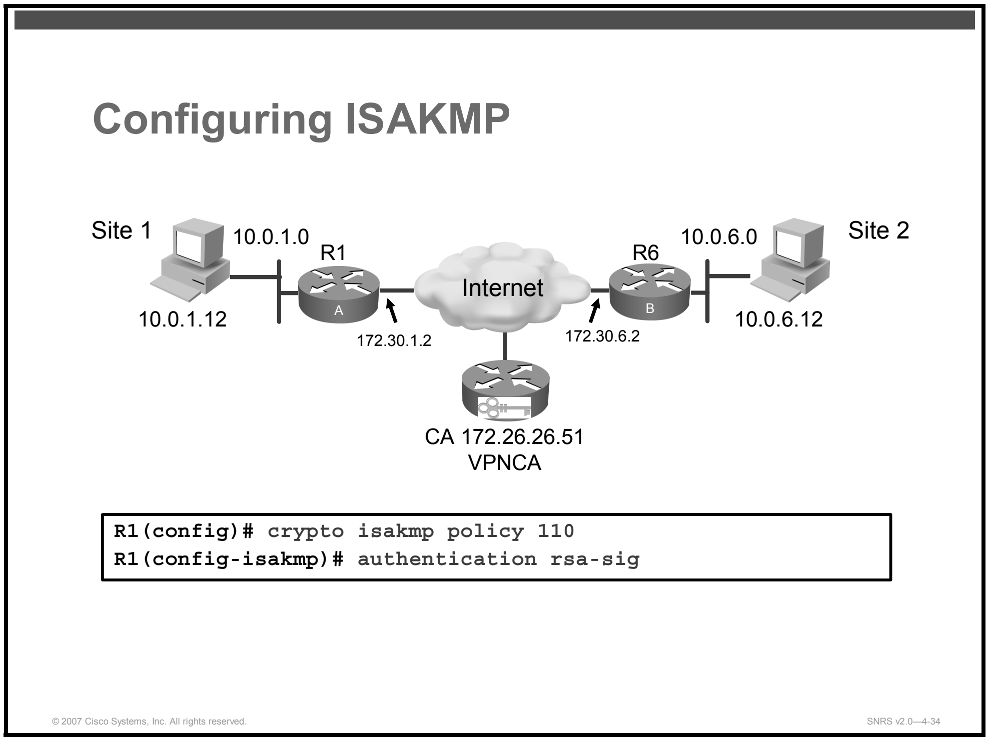
debug CA Commands

```
R1# debug crypto pki messages
R1# debug crypto pki transactions
```

Some commands are available to troubleshoot CA interoperability. You can use the **debug crypto pki messages** and the **debug crypto pki transactions** commands to assist you in finding any issues related to CA operations.

Configuring ISAKMP

This section describes how to configure ISAKMP to use RSA signatures for authentication.



Configuring ISAKMP

```
R1(config)# crypto isakmp policy 110
R1(config-isakmp)# authentication rsa-sig
```

SNRS v2.0—4-34

You will configure ISAKMP exactly in the same manner as you did with pre-shared keys, except that you will use RSA signatures for authentication instead of pre-shared keys. This figure shows an example of the way that you would configure the ISAKMP policy using RSA signatures.

Use these steps to configure ISAKMP to use RSA signatures:

Step 1 Identify the policy to create and enter the ISAKMP configuration command mode. (Each policy is uniquely identified by the priority number that you assign.)

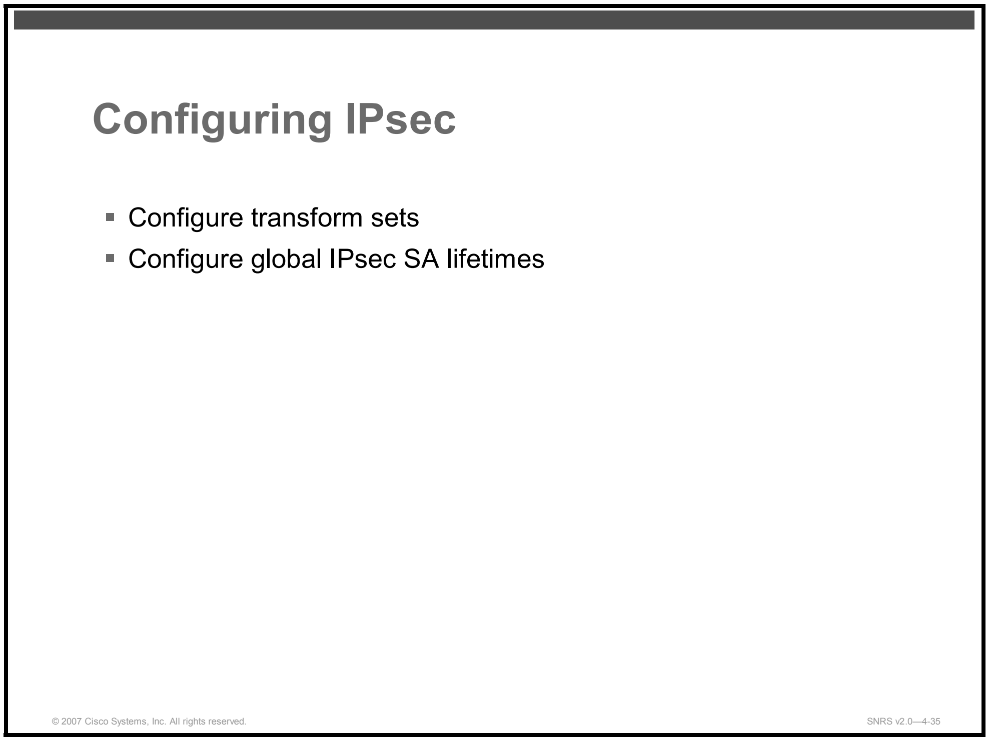
```
router(config)# crypto isakmp policy priority
```

Step 2 Specify the authentication method to use RSA signatures where an IPsec peer signs the message interchange data with the remote peer using its private key, and the remote peer uses the initiator public key to verify the digital signature. Typically, the public key is exchanged via messages containing an X.509 version 3 (X.509v3) certificate. This certificate provides a level of assurance that the identity of a peer (as represented in the certificate) is associated with a particular public key.

```
router(config-isakmp)# authentication rsa-sig
```

Configuring IPsec

This section describes how to configure an IPsec policy on a router.



Configuring IPsec

- Configure transform sets
- Configure global IPsec SA lifetimes

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-35

You will configure your IPsec policy exactly in the same manner as you did with pre-shared keys.

Use these steps to configure IPsec on the router:

Step 1 Define a transform set.

```
router(config)# crypto ipsec transform-set <transform-set-
name> transform1 [transform2[transform3]]
```

Step 2 (Optional) Change the mode associated with the transform set. The transport mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. The default mode is tunnel.

```
router(cfg-crypto-trans)# mode [transport | tunnel]
```

Step 3 (Optional) Configure global IPsec SA lifetimes.

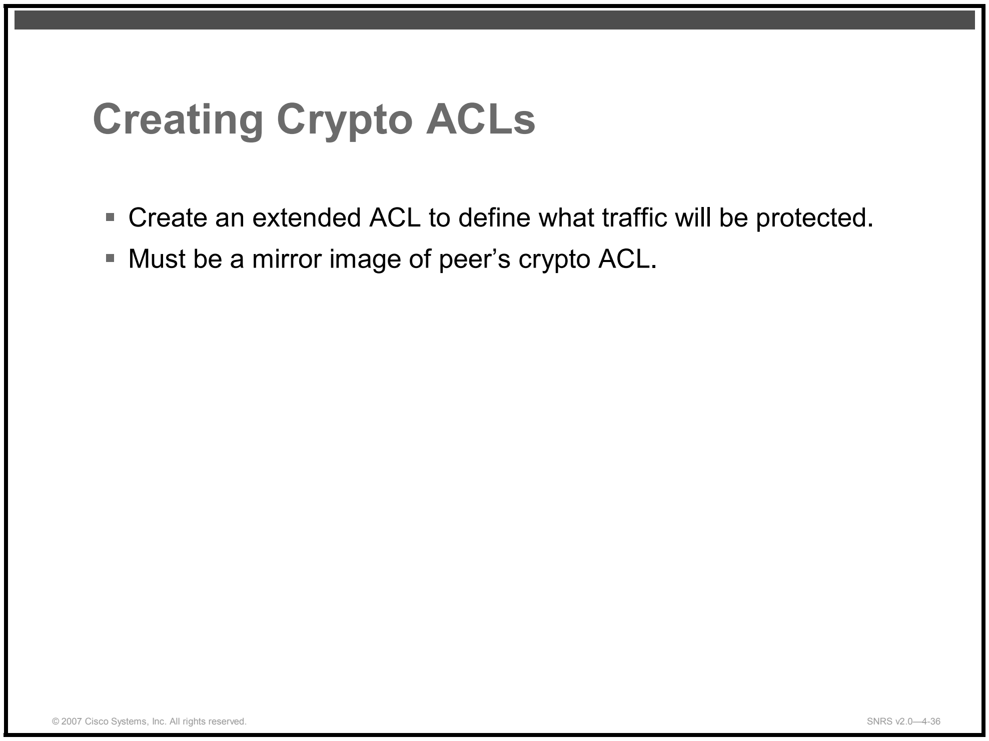
```
router(config)# crypto ipsec security-association lifetime
seconds
```

Or

```
router(config)# crypto ipsec security-association lifetime
kilobytes
```

Creating Crypto ACLs

This section describes how to create crypto ACLs.



Creating Crypto ACLs

- Create an extended ACL to define what traffic will be protected.
- Must be a mirror image of peer's crypto ACL.

SNRS v2.0—4-36

As with the previous configuration, configuring crypto ACLs for digital signatures is the same as with pre-shared keys.

Complete these steps to configure your crypto ACL:

Step 1 Create an extended ACL.

```
router(config)# ip access-list extended <name>
```

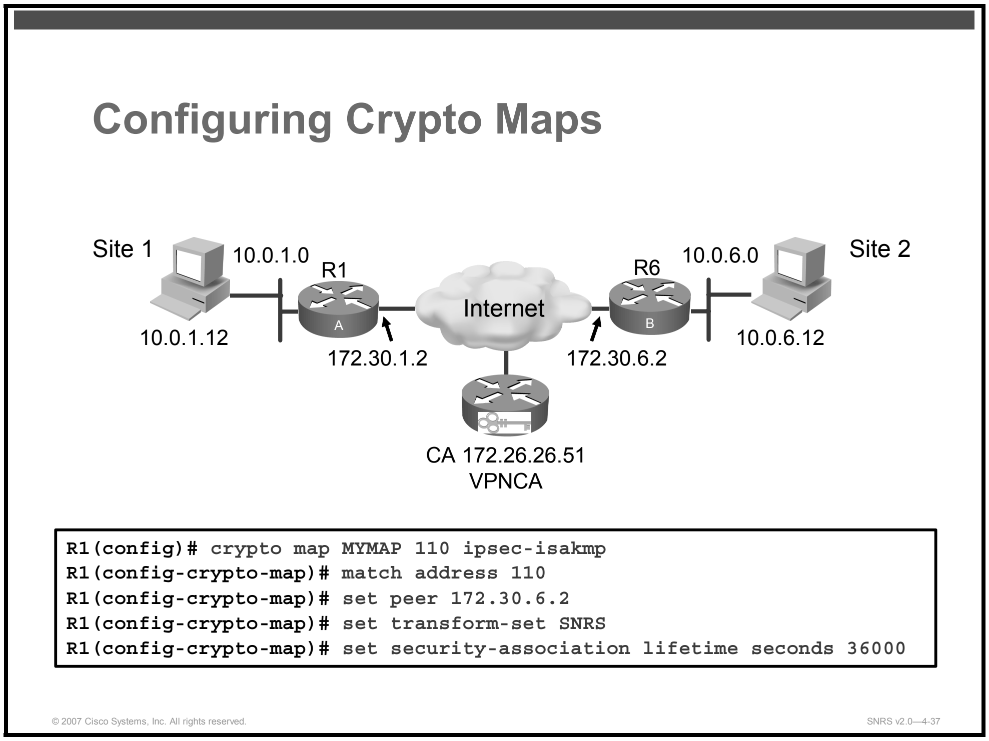
Step 2 Define which traffic is to be protected.

```
router(config-ext-nacl) permit protocol source source-wildcard
destination destination-wildcard
```

Step 3 Configure the mirror image of this ACL on the peer router.

Configuring Crypto Maps

This section describes how to create and configure IPsec crypto maps.



Configuring Crypto Maps

```
R1(config)# crypto map MYMAP 110 ipsec-isakmp
R1(config-crypto-map)# match address 110
R1(config-crypto-map)# set peer 172.30.6.2
R1(config-crypto-map)# set transform-set SNRS
R1(config-crypto-map)# set security-association lifetime seconds 36000
```

SNRS v2.0—4-37

You will configure crypto maps just as you did when using pre-shared keys.

Complete these steps to create a crypto map:

Step 1 Name the crypto map to create and give it a priority, specify ISAKMP SAs, and enter crypto map configuration mode.

Step 2 Specify an IPsec crypto ACL to match.

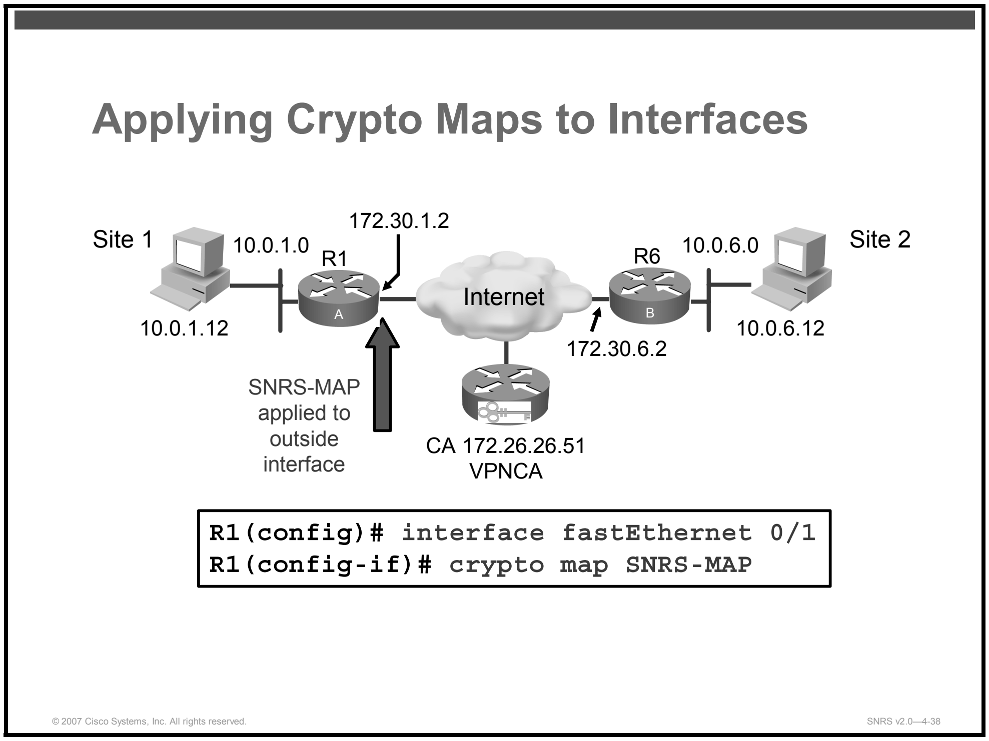
Step 3 Specify the remote IPsec peer (or peers).

Step 4 Specify which transform set should be used.

Step 5 (Optional) Set SA lifetimes.

Applying Crypto Maps to Interface

This section describes how to apply crypto maps to an interface.



Applying Crypto Maps to Interfaces

```
R1(config)# interface fastEthernet 0/1
R1(config-if)# crypto map SNRS-MAP
```

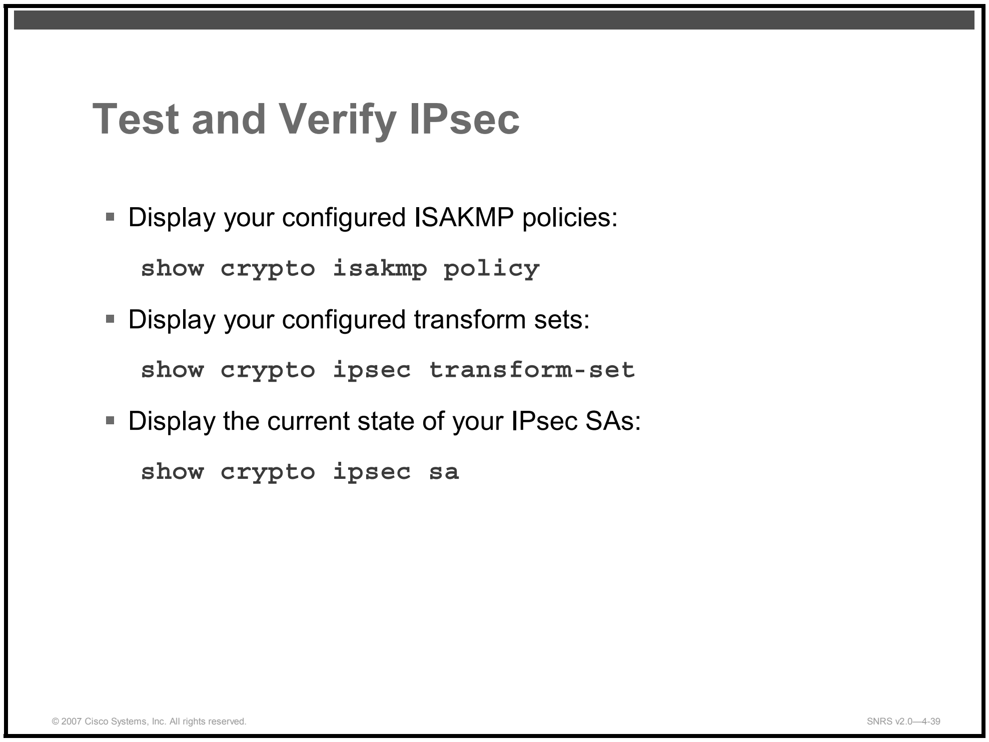
You need to apply a crypto map set to each interface through which IPsec traffic will flow just as you did with pre-shared keys.

Use the **crypto map** *map-name* command in interface configuration mode to apply the crypto map to an interface.

The example above shows the crypto map named SNRS-MAP being applied to the outside interface of R1.

Testing and Verifying IPsec

This topic describes the commands used to test and verify IPsec configurations.



Test and Verify IPsec

- Display your configured ISAKMP policies:

```
show crypto isakmp policy
```

- Display your configured transform sets:

```
show crypto ipsec transform-set
```

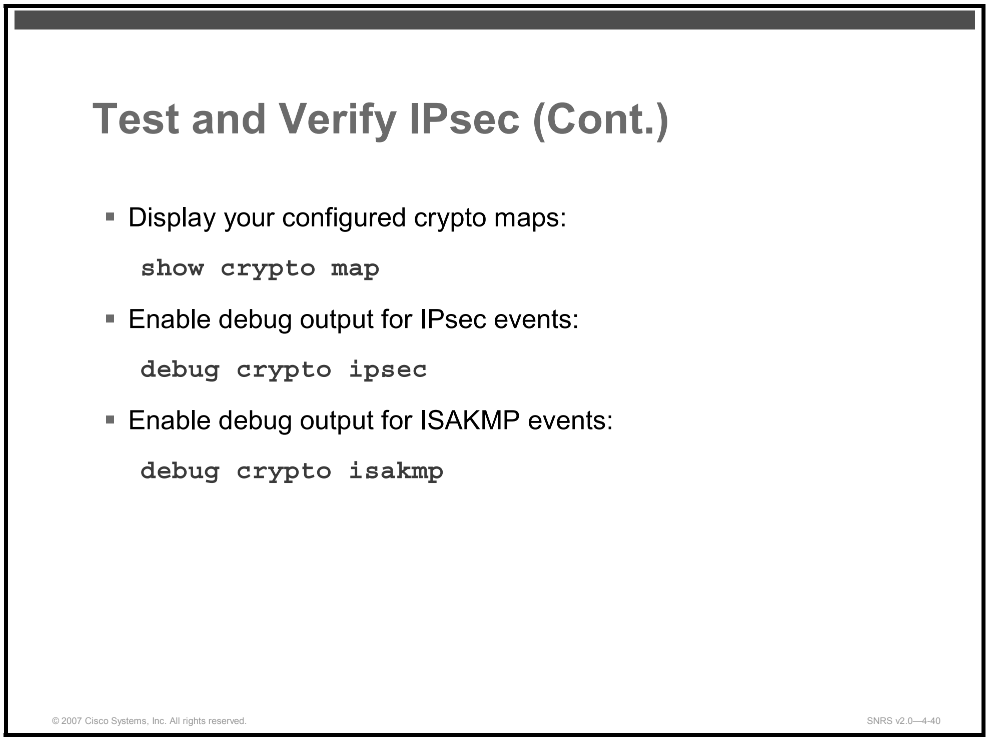
- Display the current state of your IPsec SAs:

```
show crypto ipsec sa
```

SNRS v2.0—4-39

There are several commands available to test and verify IPsec site-to-site configurations. Just as you did with pre-shared keys, you can perform the following actions to test and verify that you have correctly configured the site-to-site VPN:

- Display your configured ISAKMP policies using the **show crypto isakmp policy** command
- Display your configured transform sets using the **show crypto ipsec transform-set** command
- Display the current state of your IPsec SAs with the **show crypto ipsec sa** and **show crypto isakmp sa** commands



Test and Verify IPsec (Cont.)

- Display your configured crypto maps:

```
show crypto map
```

- Enable debug output for IPsec events:

```
debug crypto ipsec
```

- Enable debug output for ISAKMP events:

```
debug crypto isakmp
```

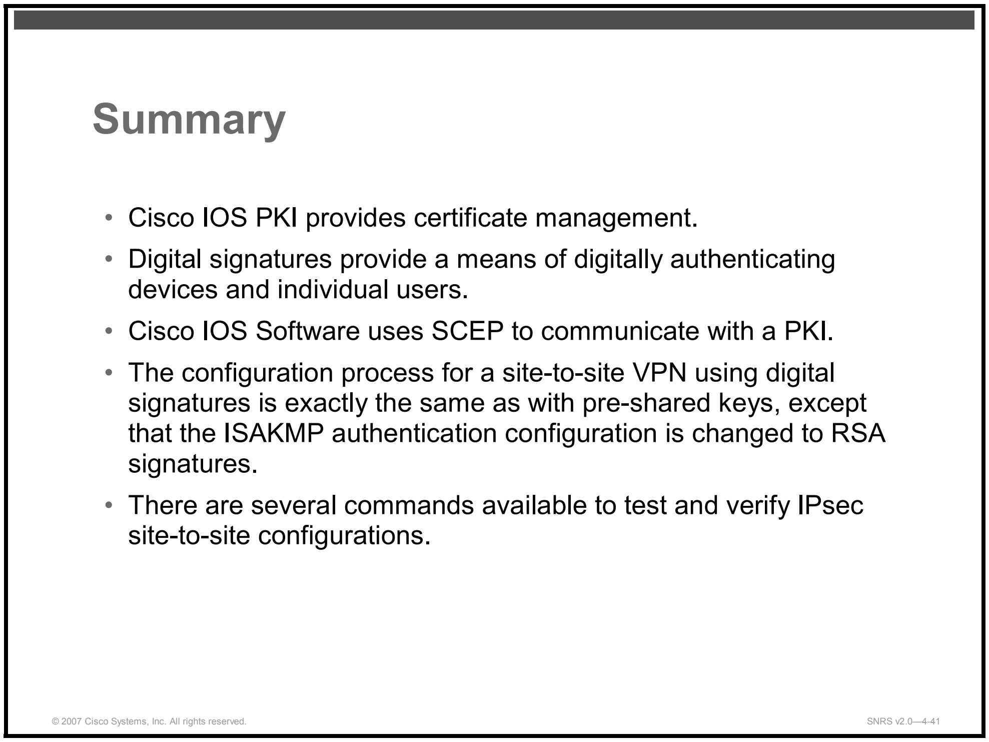
SNRS v2.0—4-40

Just as you did with pre-shared keys, you can perform the following actions to test and verify that you have correctly configured the site-to-site VPN:

- View your configured crypto maps with the **show crypto map** command
- Debug ISAKMP and IPsec traffic through the Cisco IOS Software with the **debug crypto ipsec** and **debug crypto isakmp** commands

Summary

This topic summarizes the key points that were discussed in this lesson.



Summary

- Cisco IOS PKI provides certificate management.
- Digital signatures provide a means of digitally authenticating devices and individual users.
- Cisco IOS Software uses SCEP to communicate with a PKI.
- The configuration process for a site-to-site VPN using digital signatures is exactly the same as with pre-shared keys, except that the ISAKMP authentication configuration is changed to RSA signatures.
- There are several commands available to test and verify IPsec site-to-site configurations.

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-41

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4.* http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_book09186a008043360a.html.
- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4T.* http://www.cisco.com/en/US/partner/products/ps6441/products_configuration_guide_book09186a008049e249.html.
- *Deploying Cisco IOS Security with a Public-Key Infrastructure* http://www.cisco.com/en/US/partner/tech/tk1132/technologies_white_paper09186a00800e79cb.shtml

Lesson 5

Configuring GRE Tunnels

Overview

A problem with IP Security (IPsec) tunnels is that they only work with IP packets. This means that you cannot configure a dynamic routing protocol to run over the IPsec tunnel. This lesson will introduce you to the generic routing encapsulation (GRE) protocol and how to configure an IPsec/GRE tunnel.

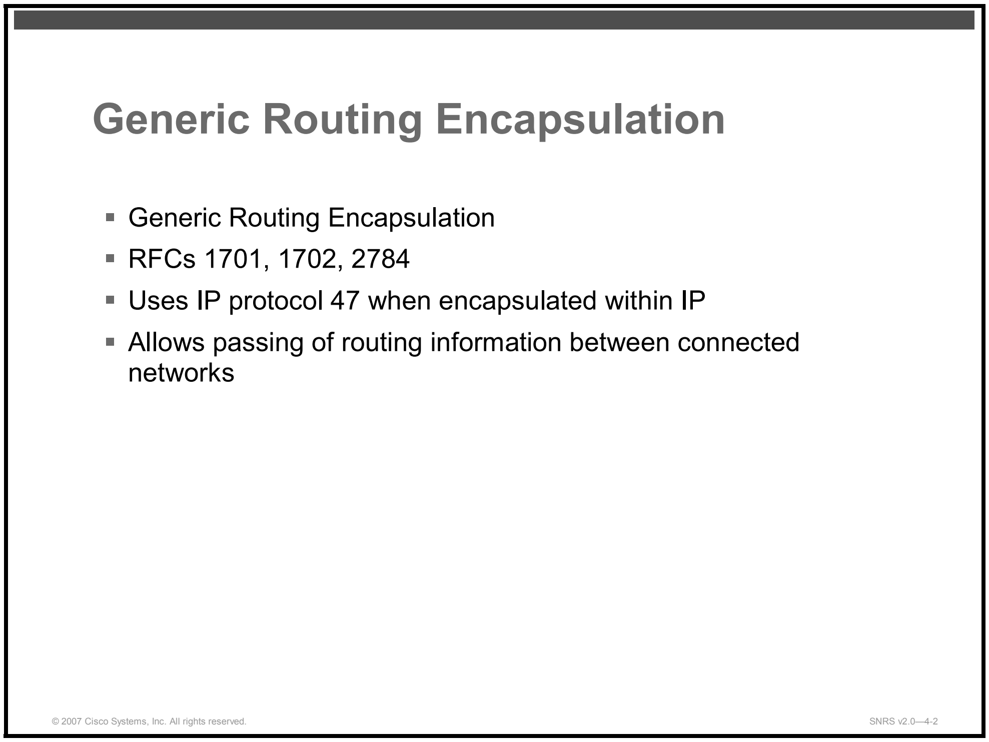
Objectives

Upon completing this lesson, you will be able to configure a GRE tunnel. This ability includes being able to meet these objectives:

- Describe GRE tunnels
- Describe how to deploy a GRE tunnel
- Describe how to configure a GRE tunnel
- Describe how to verify a GRE tunnel interface configuration
- Describe how to configure a GRE tunnel with IPsec encryption

Examining GRE Tunnels

This topic describes GRE tunnels.



Generic Routing Encapsulation

- Generic Routing Encapsulation
- RFCs 1701, 1702, 2784
- Uses IP protocol 47 when encapsulated within IP
- Allows passing of routing information between connected networks

SNRS v2.0—4-2

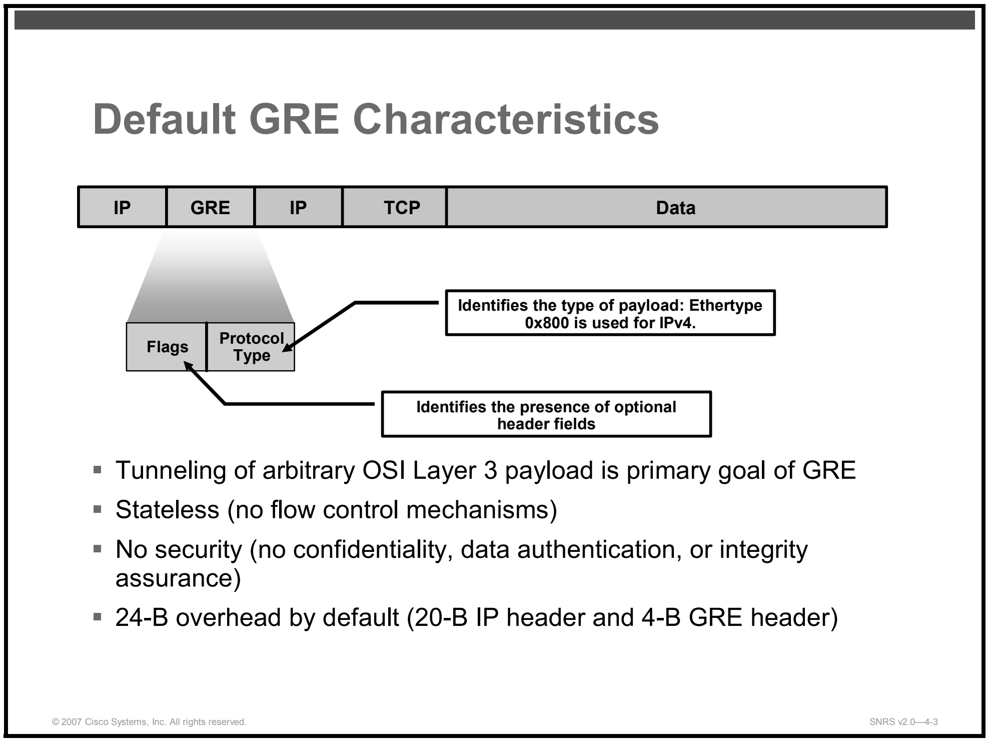
GRE is a tunneling protocol designed for encapsulation of arbitrary kinds of network layer packets inside arbitrary kinds of network layer packets as defined in RFCs 1701, 1702, and 2784. (See the References subtopic for the titles of these RFCs). RFC 1702 deals with GRE over IP version 4 (IPv4) networks. GRE was developed by Cisco Systems and can encapsulate a wide variety of protocol packet types inside IP tunnels.

Tunneling provides a way to encapsulate packets inside a transport protocol. Tunneling is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interface is not tied to specific passenger or transport protocols, but rather, it is an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. Because tunnels are point-to-point links, you must configure a separate tunnel for each link.

Tunneling allows for the encryption and transportation of multiprotocol traffic across the virtual private network (VPN) because the tunneled packets appear to the IP network as an IP unicast frame between the tunnel endpoints. If all connectivity must go through the home gateway router, tunnels also enable the use of private network addressing across a service provider backbone without the need for running the Network Address Translation (NAT) feature.

GRE uses IP protocol 47, and can be used in conjunction with IPsec VPNs to allow passing of routing information between connected networks.

As an alternative to IPsec only, traffic to be encrypted could be forwarded onto a GRE interface, which would be configured to use IPsec encryption. Packets forwarded by the GRE interface would be encapsulated and routed out onto the physical interface. GRE is capable of handling the transportation of multiprotocol and IP multicast traffic between two sites that only have IP unicast connectivity.



Default GRE Characteristics

IP	GRE	IP	TCP	Data

Flags	Protocol Type

Identifies the type of payload: Ethertype 0x800 is used for IPv4.

Identifies the presence of optional header fields

- Tunneling of arbitrary OSI Layer 3 payload is primary goal of GRE
- Stateless (no flow control mechanisms)
- No security (no confidentiality, data authentication, or integrity assurance)
- 24-B overhead by default (20-B IP header and 4-B GRE header)

SNRS v2.0—4-3

GRE encapsulation uses a protocol type field in the GRE header to support the encapsulation of any Open Systems Interconnection (OSI) Layer 3 protocol.

GRE itself is completely stateless—it does not include any flow control mechanisms by default.

GRE also does not include any strong security mechanisms to protect its payload.

The GRE header together with the tunneling IP header creates at least 24 bytes (B) of additional overhead for tunneled packets.

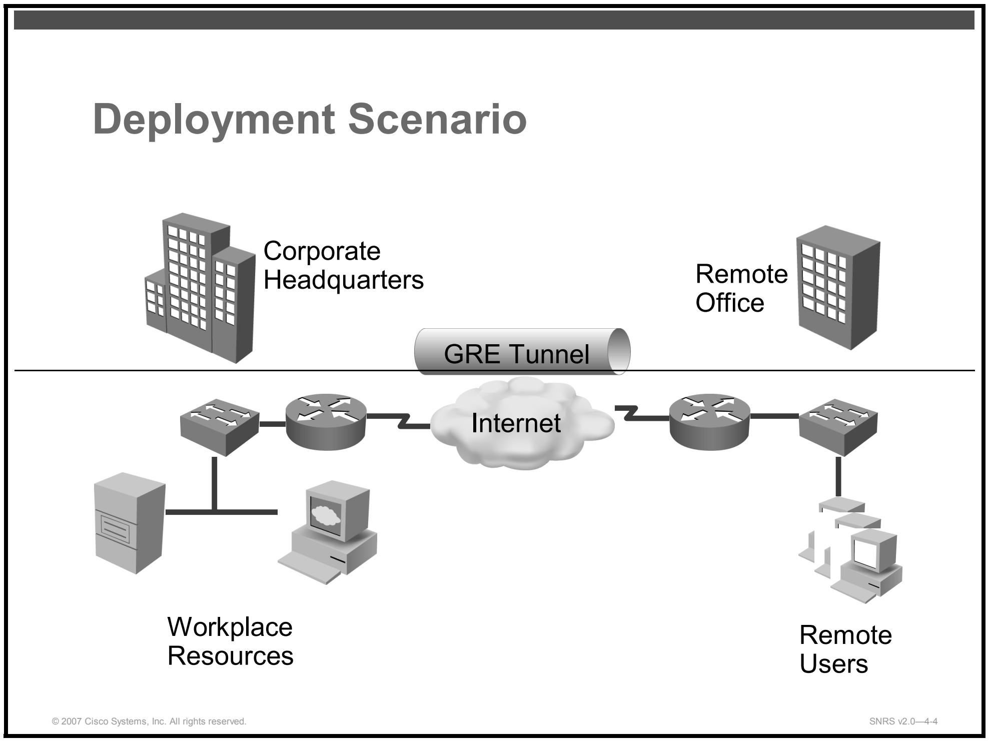
This figure illustrates some tunneling terminology and concepts. In general, a network layer packet, called the payload packet, is encapsulated in a GRE packet, which may also include source route information. The resulting GRE packet is then encapsulated in some other network layer protocol, called the delivery protocol, and then forwarded.

The three primary components of tunneling are as follows:

- Passenger or payload protocol, which is the protocol that you are encapsulating (AppleTalk, Banyan Virtual Integrated Network Service [VINES], Connectionless Network Service [CLNS], DECnet, IP, or Internetwork Packet Exchange [IPX])
- Carrier or encapsulation protocol, such as the GRE protocol or IPsec protocol
- Transport or delivery protocol, such as IP (which is the protocol used to carry the encapsulated protocol

Deploying GRE

The figure represents a typical GRE deployment scenario.



Deployment Scenario

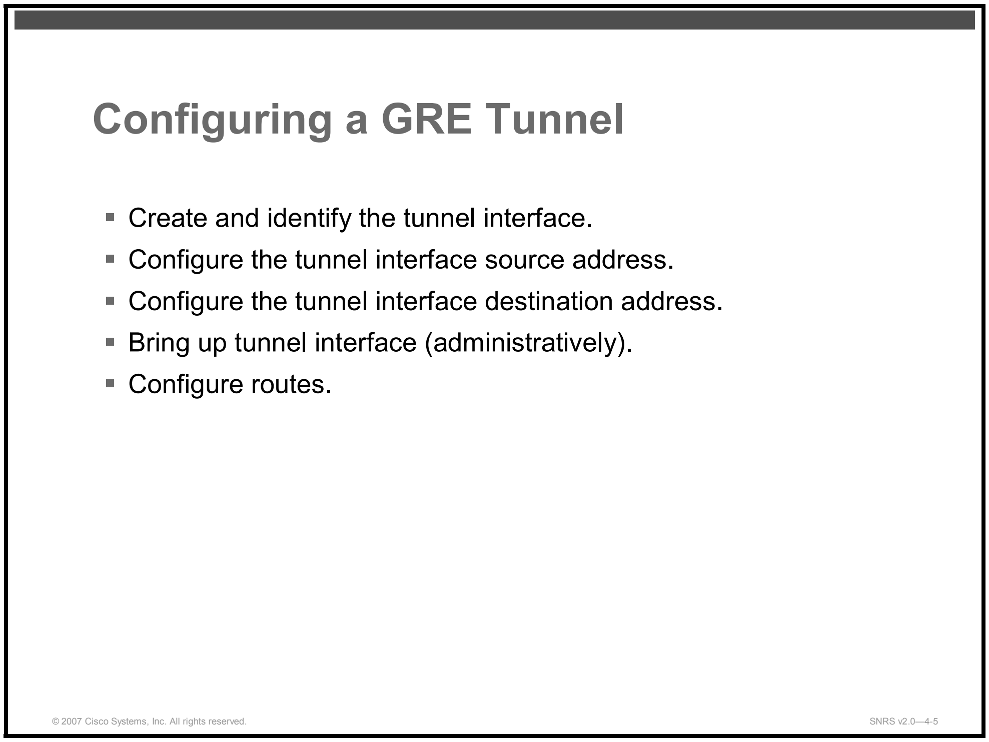
SNRS v2.0—4-4

This figure shows a headquarters network providing a remote office access to the corporate intranet. In this scenario, the headquarters and remote office are connected through a GRE tunnel that is established over an IP infrastructure (the Internet). Employees in the remote office are able to access internal, private web pages and perform various IP-based network tasks.

GRE can be used in conjunction with IPsec to pass routing updates between sites on an IPsec VPN. GRE encapsulates the cleartext packet; then, IPsec (in transport mode or tunnel mode) encrypts the packet. This packet flow of IPsec over GRE enables routing updates, which are generally multicast, to be passed over an encrypted link. IPsec alone cannot achieve this, because it does not support multicast. The importance of using tunnels in a VPN environment is based on the fact that IPsec encryption only works on IP unicast frames.

Configuring a GRE Tunnel

This topic describes how to configure a GRE tunnel.



Configuring a GRE Tunnel

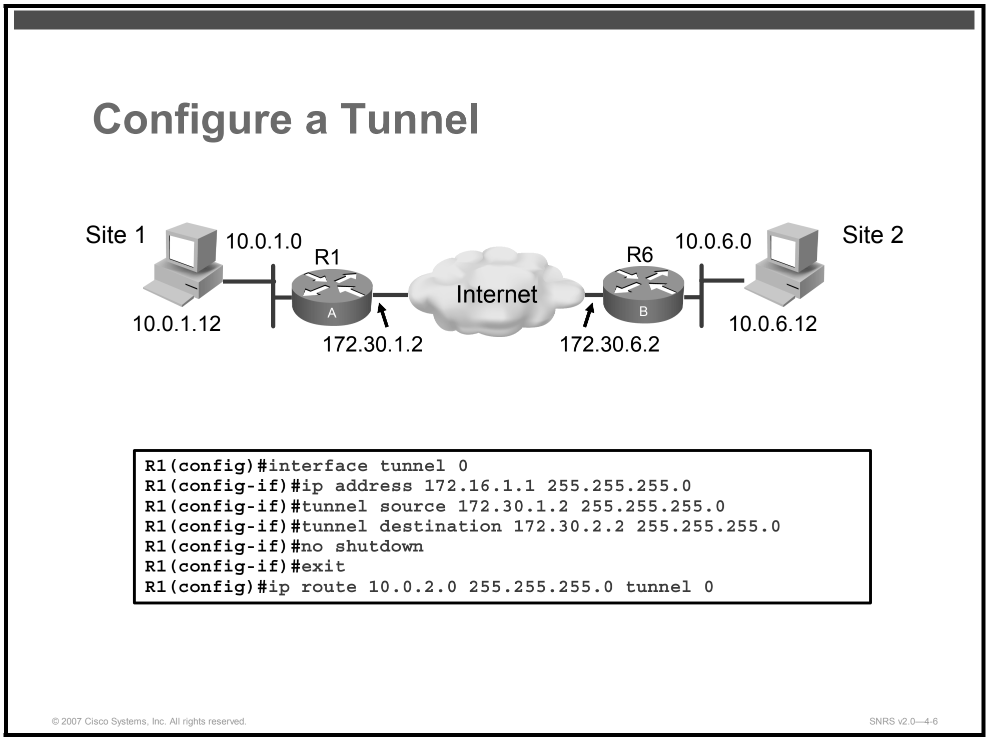
- Create and identify the tunnel interface.
- Configure the tunnel interface source address.
- Configure the tunnel interface destination address.
- Bring up tunnel interface (administratively).
- Configure routes.

To configure a GRE tunnel between the headquarters and remote office routers, you must configure a tunnel interface, source, and destination on the local and remote office routers.

When configuring GRE, you must have only Cisco routers or access servers at both ends of the tunnel connection.

Configuration Example

This figure is an example of a basic GRE configuration.



Configure a Tunnel

```
R1(config)#interface tunnel 0
R1(config-if)#ip address 172.16.1.1 255.255.255.0
R1(config-if)#tunnel source 172.30.1.2 255.255.255.0
R1(config-if)#tunnel destination 172.30.2.2 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip route 10.0.2.0 255.255.255.0 tunnel 0
```

SNRS v2.0—4-6

To configure a basic GRE tunnel, follow these steps:

Step 1 Specify a tunnel interface number and enter interface configuration mode.

```
router(config)# interface tunnel number
```

Step 2 Configure an IP address and subnet mask on the tunnel interface.

```
router(config-if)# ip address ip-address net-mask
```

Step 3 Specify the tunnel interface source address and subnet mask.

```
router(config-if)# tunnel source source-ip source-net-mask
```

Syntax Description

source-ip	This is the address of the local router interface
source-net-mask	This is the netmask of the local router interface

Caution .The tunnel source and destination should be on the same subnet.

Step 4 Specify the tunnel interface destination address.

```
router(config-if)# tunnel destination dest-ip dest-net-mask
```

Syntax Description

dest-ip	This is the address of the remote router interface
dest-net-mask	This is the netmask of the remote router interface

Step 5 Bring up the tunnel interface.

```
router(config-if)# no shutdown
```

Step 6 Exit back to global configuration mode.

```
router(config-if)# exit
```

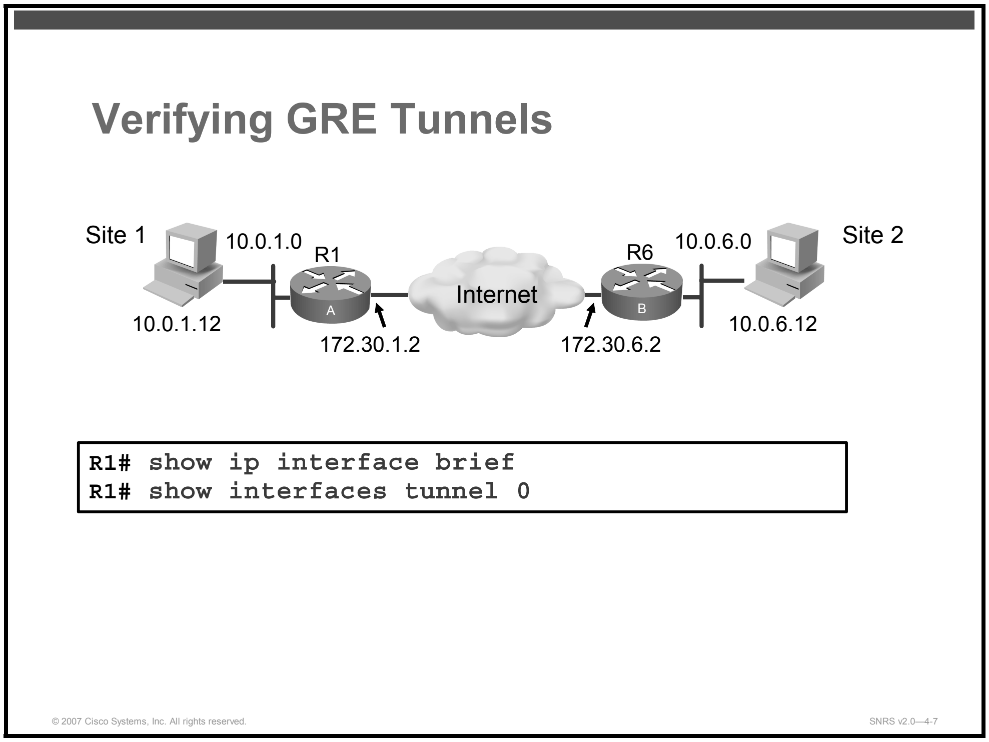
Step 7 Configure traffic from the remote office network through the tunnel.

```
router(config)# ip route remote-network remote-mask tunnel number
```

GRE tunnels can be used alone or with IPsec to encrypt the traffic passing through the tunnel. This example is shown in not using IPsec.

Verifying GRE Tunnels

This topic describes how to verify a GRE tunnel interface configuration.



Verifying GRE Tunnels

```
R1# show ip interface brief
R1# show interfaces tunnel 0
```

SNRS v2.0—4-7

There are several commands used to monitor and troubleshoot GRE tunnels.

- To check if the tunnel interface is up or down, use the **show ip interface brief** command as follows:

```
router# show ip interface brief
Interface                  IP-Address      OK? Method Status          Protocol
FastEthernet0/0            10.0.1.2        YES NVRAM  up                    up


FastEthernet0/1            172.30.1.2      YES NVRAM  up                    up


Tunnel0                    172.16.1.1      YES manual up                    up
```

- To verify tunnel interface configuration, use the **show interfaces tunnel** command as follows:

```
        router# show interfaces tunnel number [accounting]
```

Syntax Description

number	Number of the tunnel
accounting	Displays the number of packets of each protocol type that have been sent through the interface

The output should be similar to the following:

```
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.2, destination 172.30.2.2
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

- Verify connectivity by pinging the other side of the tunnel as follows:

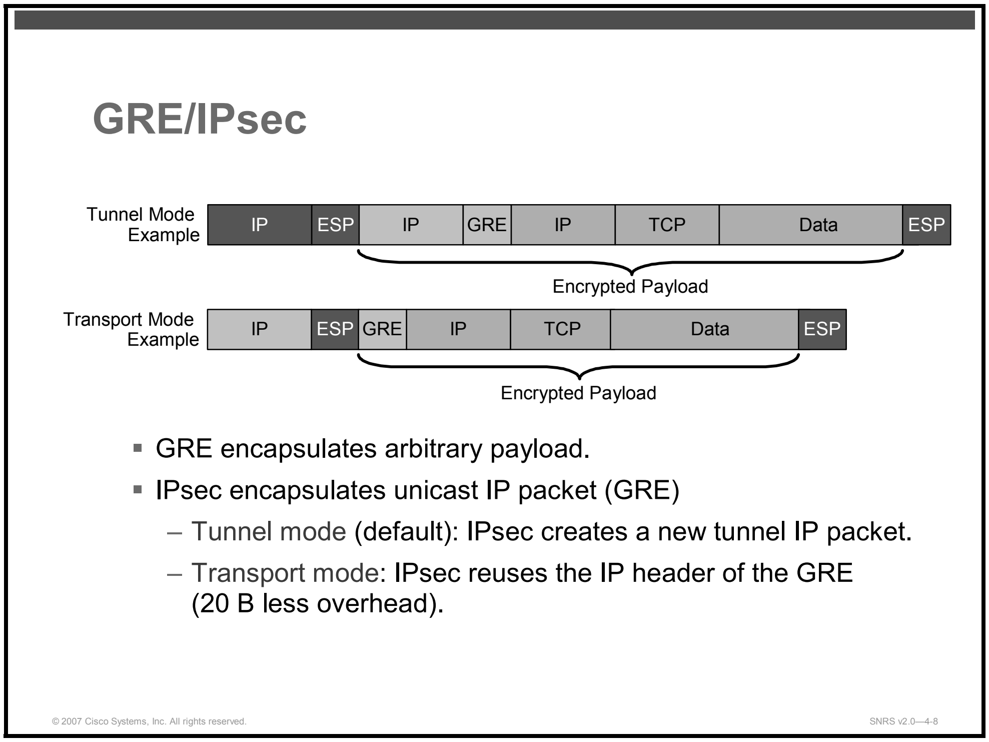
```
routerA# ping 172.30.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

- Then ping a host on the remote subnet as follows:

```
router1# ping 10.0.6.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Configuring GRE Tunnels and Encryption

This topic describes how to configure a GRE tunnel with IPsec encryption.



GRE/IPsec

Tunnel Mode Example	IP	ESP	IP	GRE	IP	TCP	Data	ESP

Encrypted Payload

Transport Mode Example	IP	ESP	GRE	IP	TCP	Data	ESP

Encrypted Payload

- GRE encapsulates arbitrary payload.
- IPsec encapsulates unicast IP packet (GRE)
  - Tunnel mode (default): IPsec creates a new tunnel IP packet.
  - Transport mode: IPsec reuses the IP header of the GRE (20 B less overhead).

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-8

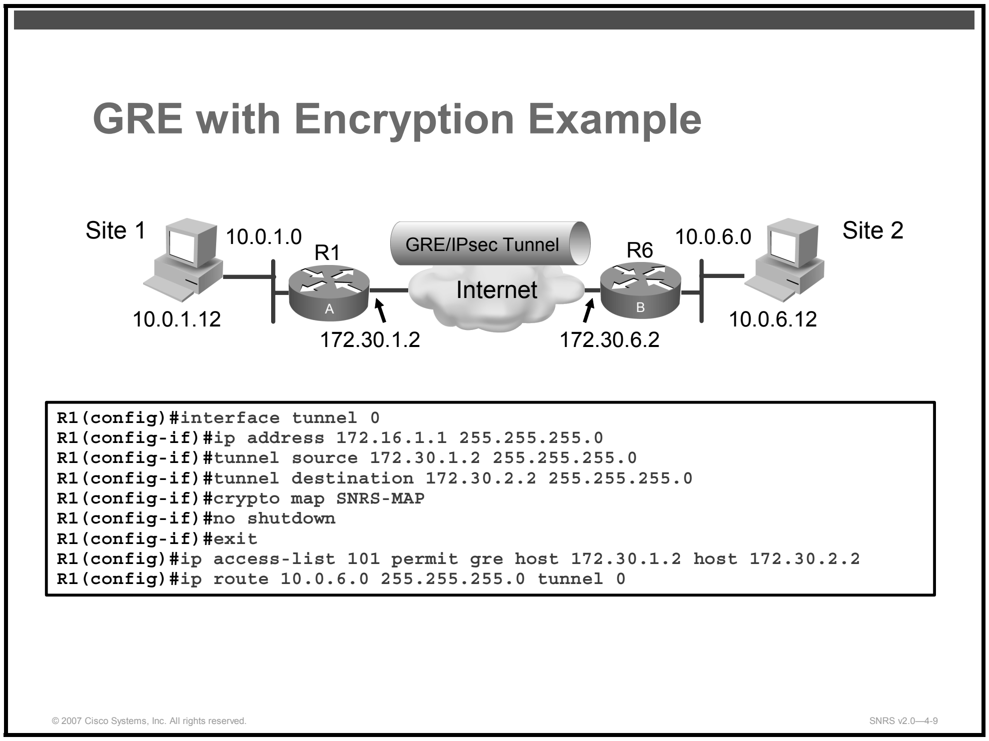
When GRE tunnel endpoints are located at the encrypting routers of the peer, you can configure encryption so that all traffic through the GRE tunnel is encrypted.

Note You cannot selectively encrypt GRE tunnel traffic: either all the GRE tunnel traffic is encrypted or no GRE tunnel traffic is encrypted.

To configure encryption with GRE tunnels, you will perform the same basic tasks described in the lesson “Implementing IPsec VPNs Using Pre-Shared Keys" plus some additional steps.

Encryption ACLs and GRE

When using IPsec with GRE, the access control list (ACL) for encrypting traffic does not define the traffic to be protected; instead, it should allow GRE between the source and destination of the GRE tunnel. Without a further ACL on the tunnel interface, this configuration will allow for all packets forwarded to the GRE tunnel to get encrypted.



GRE with Encryption Example

```
R1(config)#interface tunnel 0
R1(config-if)#ip address 172.16.1.1 255.255.255.0
R1(config-if)#tunnel source 172.30.1.2 255.255.255.0
R1(config-if)#tunnel destination 172.30.2.2 255.255.255.0
R1(config-if)#crypto map SNRS-MAP
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip access-list 101 permit gre host 172.30.1.2 host 172.30.2.2
R1(config)#ip route 10.0.6.0 255.255.255.0 tunnel 0
```

SNRS v2.0—4-9

Encrypting GRE Tunnel Traffic

To encrypt only traffic through the GRE tunnel, follow these additional instructions:

- When you set up your encryption ACL, the list should contain only one criteria statement. In this one statement, specify “gre” as the protocol, specify the tunnel source address as the source, and specify the tunnel destination address as the destination.
- Apply the crypto map to both the physical interface and to the tunnel interface.

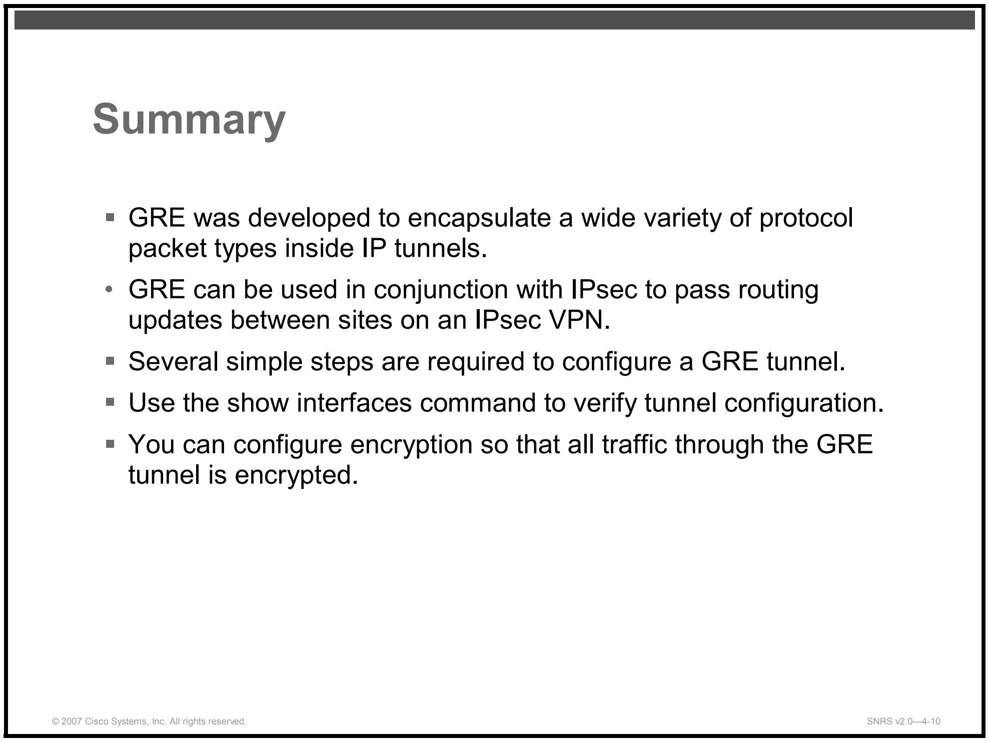
Note	Without GRE tunnels, you only had to apply the crypto map to the physical interface.

- In addition to creating a tunnel interface, the ACL used for the crypto map must be modified to only permit the GRE traffic between the outside interfaces of both peers.

Repeat this at both ends of the GRE tunnel.

Summary

This topic summarizes the key points that were discussed in this lesson.



Summary

- GRE was developed to encapsulate a wide variety of protocol packet types inside IP tunnels.
- GRE can be used in conjunction with IPsec to pass routing updates between sites on an IPsec VPN.
- Several simple steps are required to configure a GRE tunnel.
- Use the show interfaces command to verify tunnel configuration.
- You can configure encryption so that all traffic through the GRE tunnel is encrypted.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-10

References

For additional information, refer to these resources:

- RFC 1701, *Generic Routing Encapsulation (GRE)*
- RFC 1702, *Generic Routing Encapsulation over IPv4 Networks*
- RFC 2784, *Generic Routing Encapsulation (GRE)*

Lesson 6

Configuring a DMVPN

Overview

Dynamic Multipoint Virtual Private Networks (DMVPNs) make the configuration of multiple sites to a company headquarters relatively easy. In this lesson, you will learn to integrate multipoint generic routing encapsulation (mGRE), Next Hop Resolution Protocol (NHRP), and IP Security (IPsec) profiles to create a DMVPN.

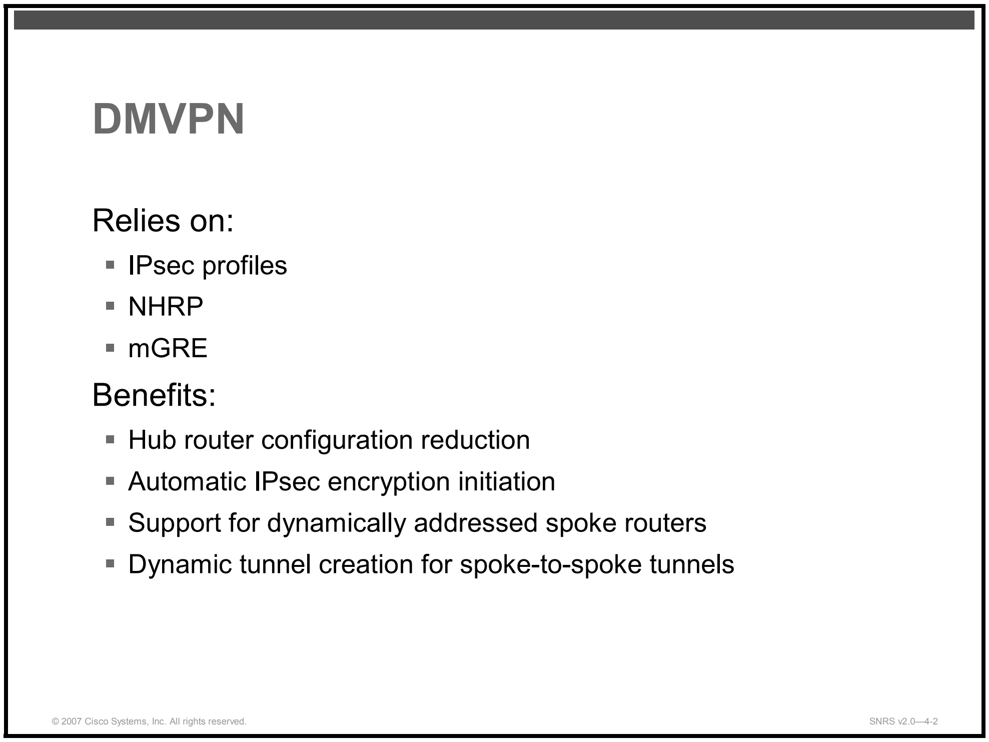
Objectives

Upon completing this lesson, you will be able to configure a DMVPN. This ability includes being able to meet these objectives:

- Describe the overall features, operation, and prerequisites for DMVPN
- Describe the tasks required to configure a DMVPN
- Describe how to create ISAKMP policies and IPsec transforms for use with a DMVPN
- Describe how to create an IPsec profile to use with tunnel protection mode
- Describe routing protocol issues with DMVPN
- Describe the commands used to configure the hub in a spoke-to-spoke DMVPN network
- Describe the commands used to configure the spoke in a spoke-to-spoke DMVPN network
- Describe how to verify DMVPN connectivity

Dynamic Multipoint VPN

This topic describes the overall features, operation, and prerequisites for DMVPN.



DMVPN

Relies on:

- IPsec profiles
- NHRP
- mGRE

Benefits:

- Hub router configuration reduction
- Automatic IPsec encryption initiation
- Support for dynamically addressed spoke routers
- Dynamic tunnel creation for spoke-to-spoke tunnels

SNRS v2.0—4-2

The Cisco DMVPN feature allows users to better scale large and small IPsec virtual private networks (VPNs). The Cisco DMVPN feature combines mGRE tunnels, IPsec encryption, and NHRP routing to provide users an ease of configuration via crypto profiles, which override the requirement for defining static crypto maps, and dynamic discovery of tunnel endpoints.

The Cisco DMVPN feature relies on the following two technologies:

- **NHRP:** This is a client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of the each spoke. Each spoke registers its real address when it boots and queries the NHRP database for the real addresses of the destination spokes to build direct tunnels.
- **mGRE:** This allows a single generic routing encapsulation (GRE) interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.

Benefits

Here are the primary benefits of DMVPNs:

- **Hub router configuration reduction:** Currently, for each spoke router, there is a separate block of configuration lines on the hub router that define the crypto map characteristics, the crypto access control list (ACL), and the GRE tunnel interface. This feature allows users to configure a single mGRE tunnel interface, a single IPsec profile, and does not require crypto ACLs on the hub router to handle all spoke routers. Thus, the size of the configuration on the hub router remains constant even if spoke routers are added to the network.

- **Automatic IPsec encryption initiation:** GRE has the peer source and destination address configured or resolved with NHRP. Thus, this feature allows IPsec to be immediately triggered for the point-to-point GRE tunneling or when the GRE peer address is resolved via NHRP for the mGRE tunnel.
- **Support for dynamically addressed spoke routers:** When using point-to-point GRE and IPsec hub-and-spoke VPN networks, the physical interface IP address of the spoke routers must be known when configuring the hub router because the IP address must be configured as the GRE tunnel destination address. This feature allows spoke routers to have dynamic physical interface IP addresses.

Note Dynamic addressing is common for cable and DSL connections.

When the spoke router comes on line, it will send NHRP registration packets to the hub router. Within these registration packets is the current physical interface IP address of this spoke.

- **Dynamic tunnel creation for spoke-to-spoke tunnels:** This feature eliminates the need for spoke-to-spoke configuration for direct tunnels. When a spoke router wants to transmit a packet to another spoke router, it can now use NHRP to dynamically determine the required destination address of the target spoke router. (The hub router acts as the NHRP server, handling the request for the source spoke router.) The two spoke routers dynamically create an IPsec tunnel between them so that data can be directly transferred.

Prerequisites

Before an mGRE and IPsec tunnel can be established, you must define an Internet Key Exchange (IKE) policy by using the **crypto isakmp policy** command.

Restrictions

If you use the dynamic tunnel creation for spoke-to-spoke tunnels benefit of this feature, you must use IKE certificates or wildcard pre-shared keys for Internet Security Association Key Management Protocol (ISAKMP) authentication.

Note It is highly recommended that you do not use wildcard pre-shared keys, because the attacker will have access to the VPN if one spoke router is compromised.

GRE tunnel keepalives (that is, the **keepalive** command under the GRE tunnel interface) are not supported on mGRE tunnels.

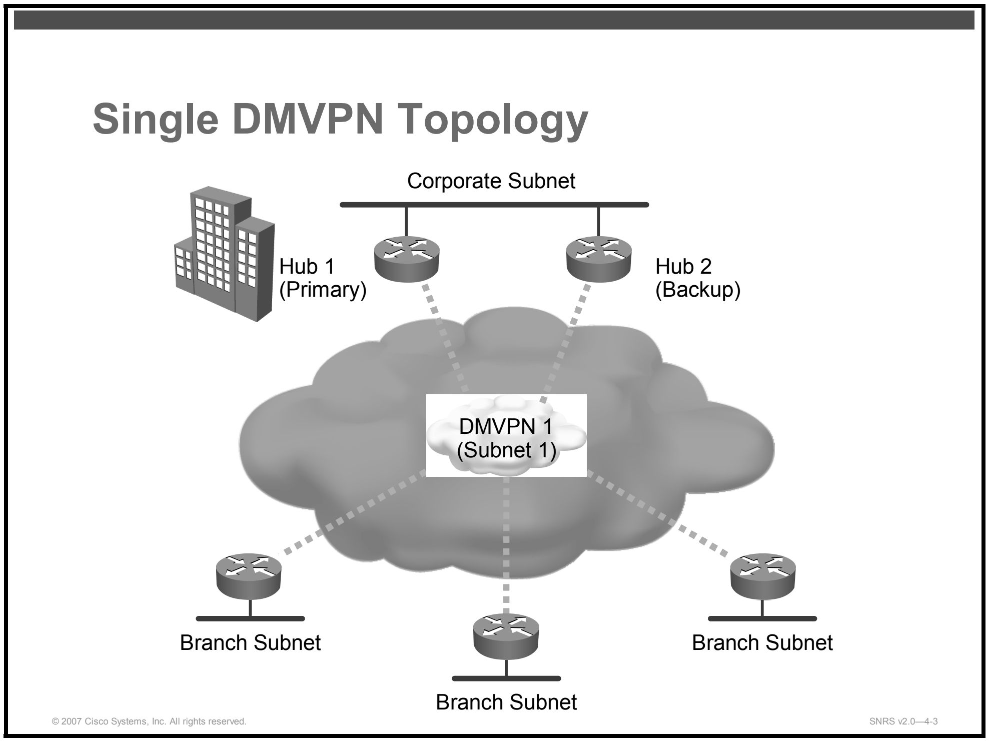
DMVPN Topologies

In a DMVPN design, the following two topologies are recommended:

- Dual hub-dual DMVPN cloud
- Dual hub-single DMVPN cloud

In both topologies, two hubs are recommended for redundancy. High availability is provided through the use of a second hub router, which may be on the same DMVPN subnet as the primary router. This is commonly referred to as a single DMVPN cloud topology. The second hub router can also service its own DMVPN subnet, which is known as a dual DMVPN cloud topology. A dual hub-single DMVPN topology is generally not recommended because it relies on mechanisms outside of the tunnel to determine the appropriate hub for failover. In contrast, hubs using dual DMVPN subnets (dual DMVPN cloud topology) rely on routing protocols running inside of the tunnel to determine path selection.

The figures represent the two topologies mentioned. The difference between the two topologies is most apparent on the branch router. With a single DMVPN subnet, the branch router has a single mGRE tunnel, and both hubs are mapped to this tunnel through an mGRE interface. In a dual DMVPN topology, the branch router has a unique tunnel pointing to a unique hub. Standard routing protocols such as Open Shortest Path First (OSPF) or Enhanced Interior Protocol (EIGRP) are used to determine the active hub.

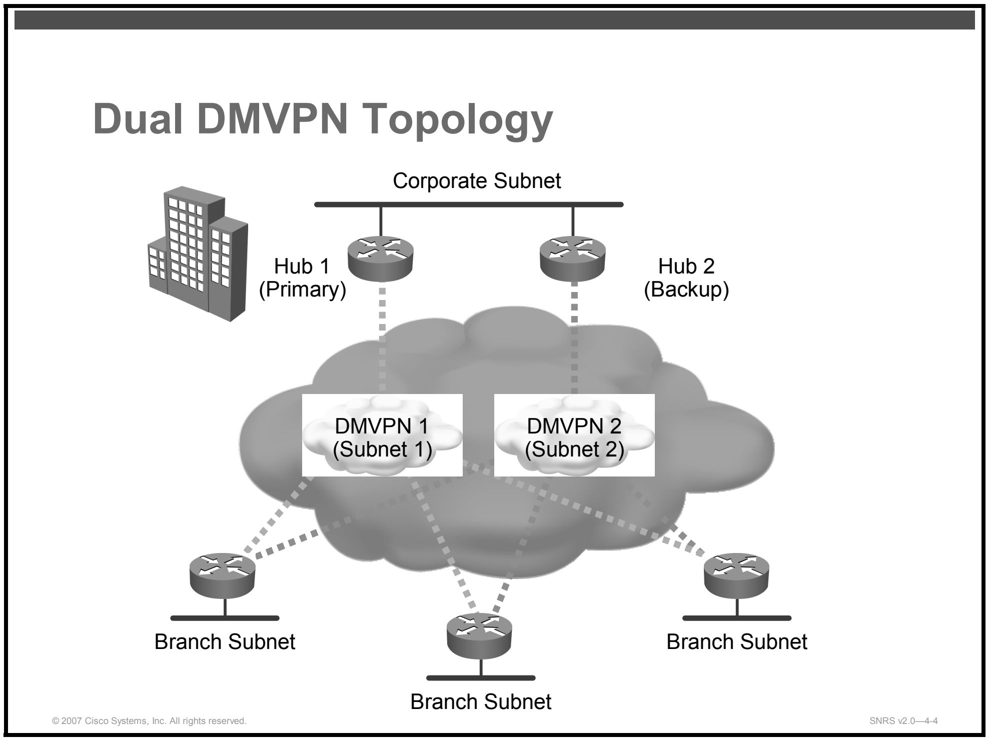


Single DMVPN Topology

SNRS v2.0—4-3

This figure represents a single DMVPN topology. In a single DMVPN cloud topology, there are two hub routers on the same DMVPN subnet. Therefore, the branch router requires an mGRE interface. Because of this mGRE interface, branch routers attempt interbranch communications if so directed by the routing table. As a result, this model should be considered a spoke-to-spoke topology. The hub-and-spoke deployment model can be configured in a single DMVPN cloud topology with only one hub router. This scenario is not recommended because there is no failover mechanism for the hub router.

A single DMVPN cloud topology with the spoke-to-spoke deployment model also contains two hub routers. The hub routers are configured similarly to the hub router configurations in the dual DMVPN cloud topology, but only one IP subnet is used. If the hubs are co-located, traffic can be load-balanced between the two hub routers. In this topology, all branch and hub mGRE interfaces are on a single subnet, which contrasts to the dual DMVPN cloud topology where there are multiple subnets each represented by a DMVPN cloud. In this scenario, there is limited control over the routing protocol, and possible asymmetric routing issues may occur.

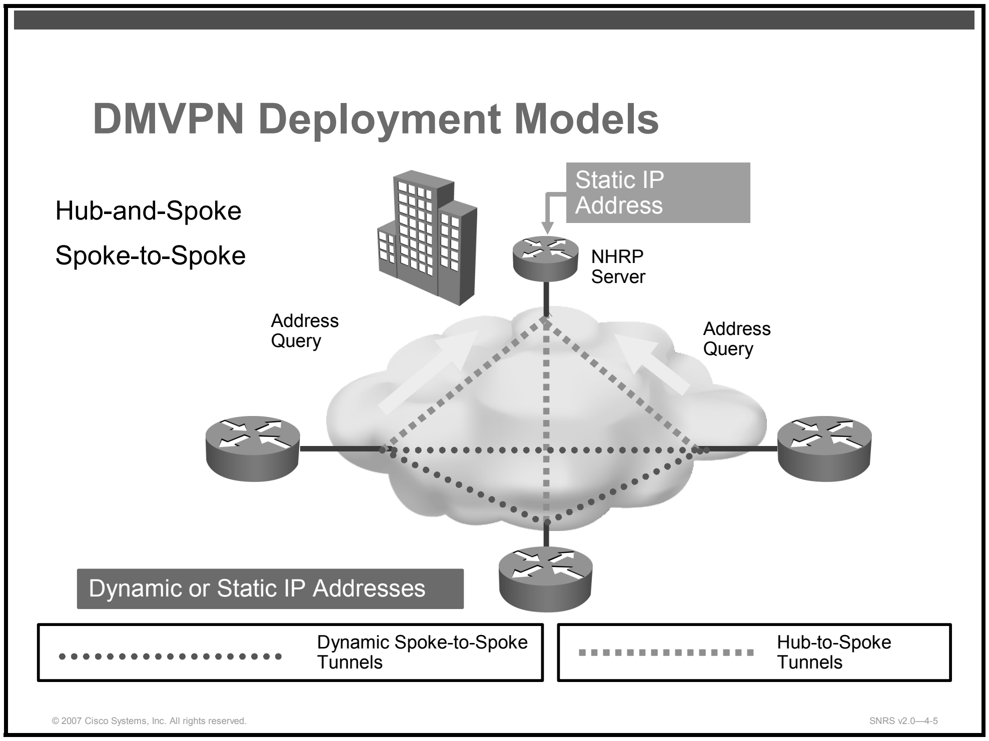


Dual DMVPN Topology

SNRS v2.0—4-4

This figure represents a dual DMVPN topology. A dual DMVPN cloud topology hub-and-spoke deployment model consists of two hub routers (hub 1 and hub 2), each with one or more mGRE tunnel interfaces that connect to all branch routers.

Each DMVPN cloud represents a unique IP subnet. One DMVPN cloud is considered the primary, which all branch traffic transits. Each branch is configured with two point-to-point GRE tunnel interfaces, with one going to each respective hub. In this deployment model, there are no tunnels between branches. Interbranch communications are provided through the hub routers. This closely matches traditional Frame Relay networks. Routing metrics are used to steer traffic to the primary hub router (hub 1).



DMVPN Deployment Models

Hub-and-Spoke

Spoke-to-Spoke

Static IP Address
NHRP Server
Address Query
Address Query
Dynamic or Static IP Addresses
•••••••••••••••••• Dynamic Spoke-to-Spoke Tunnels
■■■■■■■■■■■■■■ Hub-to-Spoke Tunnels

SNRS v2.0—4-5

A DMVPN cloud topology can support either a hub-and-spoke or spoke-to-spoke deployment model. In a hub-and-spoke deployment model, each hub contains an mGRE interface and each branch contains a point-to-point GRE interface. In a spoke-to-spoke deployment model, both the hub and the branch contain mGRE interfaces.

A DMVPN cloud is a collection of routers that is configured either with an mGRE interface or a point-to-point GRE interface (or a combination of the two) and that share the same address subnet.

A DMVPN cloud topology can support either of the following deployment models:

- **Hub-and-spoke:** The hub-and-spoke deployment model is the most common deployment model. This model is the most scalable, and predominately mimics traditional Layer 2 leased line, Frame Relay, or ATM hub-and-spoke networks. The hub is configured with an mGRE interface, and the branch with a point-to-point GRE interface.

  In this deployment model, no tunnels connect one branch to another branch. Traffic between branches passes through the hub router.

- **Spoke-to-spoke:** The spoke-to-spoke deployment model allows branches to dynamically create tunnels between other branches within the same DMVPN cloud for intercommunication. This deployment model is a fully meshed topology and requires mGRE interfaces to be configured on both the hub and all branches.

  In a spoke-to-spoke deployment model, all branch-to-branch communications transit through the hub until the dynamic spoke-to-spoke tunnel is created. The dynamic spoke-to-spoke tunnels must be within a single DMVPN cloud or subnet. It is not possible to dynamically create a spoke-to-spoke tunnel between two DMVPN clouds.

The spoke-to-spoke deployment model is very similar to the hub-and-spoke deployment model, with the exception that all GRE interfaces in the hub and the branch are mGRE interfaces. Branch routers can initiate and accept dynamic tunnels from other branch offices.

Spoke-to-Spoke DMVPN Operation

This is how a spoke-to-spoke DMVPN works:

1. Each spoke has a permanent IPsec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server.
2. When a spoke needs to send a packet to a destination (private) subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke.
3. After the originating spoke learns the peer address of the target spoke, it can initiate a dynamic IPsec tunnel to the target spoke.
4. The spoke-to-spoke tunnel is built over the mGRE interface.
5. The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets are able to bypass the hub and use the spoke-to-spoke tunnel.
6. After a preconfigured amount of inactivity on the spoke-to-spoke tunnels, the router will tear down the spoke-to-spoke tunnels to save resources.

The primary deployment model is a hub-and-spoke model in which the primary enterprise resources are located in a large central site, with a number of smaller sites or branch offices connected directly to the central site over a VPN. However, in some scenarios, a spoke-to-spoke deployment model can be used to create temporary connections between branch sites directly using IPsec encryption.

IP Addressing

Because VPNs are used for secure enterprise communications across a shared public infrastructure such as the Internet, two distinct IP address domains must be considered:

- The enterprise addressing space, sometimes referred to as the private or inside addresses
- The infrastructure addressing space, also referred to as the service provider, public, or outside addresses

In most DMVPN designs, the outside interface of the router is addressed in the infrastructure (or public) address space, assigned by the service provider. The tunnel interface belongs to the enterprise private network address space. A branch router public IP address is either a statically defined or a dynamically assigned IP address. For a hub-and-spoke deployment model, both the point-to-point GRE and crypto tunnels are sourced from the public IP address. For a spoke-to-spoke deployment model, the mGRE and crypto tunnels are also sourced from the public IP address. This address is registered with the hub router, which provides a mapping to the branch private address.

Multipoint GRE

In DMVPN designs, an mGRE interface is introduced, which serves as a one-to-many interface for the creation of multiple hub-and-spoke tunnels that work similarly to a point-to-multipoint Frame Relay interface. Unlike point-to-point GRE tunnels, the tunnel destination for an mGRE tunnel does not have to be configured. In all DMVPN designs, the hub is configured with an mGRE interface to allow the dynamic creation of tunnels for each branch connected. An mGRE interface does not require a unique tunnel interface, a unique crypto map, or a unique crypto ACL for each branch in the network. The mGRE interfaces reduce the configuration file on each hub router, which is an advantage for large-scale designs when compared to static point-to-point GRE topologies.

The deployment model chosen determines which type of GRE interface is configured on a branch router. A hub-and-spoke deployment model requires each branch to be configured with a point-to-point GRE interface. A spoke-to-spoke deployment model requires each branch to be configured with an mGRE interface.

Both point-to-point GRE and mGRE add to the size of the original data packet, including a 4-byte (B) GRE header, a 4-B mGRE tunnel key, and 20 B for an additional IP header.

The protocol header for an mGRE packet is 4 B larger than a point-to-point GRE packet. The additional 4 B constitute a tunnel key value, which is used to differentiate between different mGRE interfaces in the same router. Without a tunnel key, a router can support only one mGRE interface corresponding to one IP network. Tunnel keys allow a branch router to have a different mGRE interface corresponding to each DMVPN cloud in the network topology. A hub router can be configured as well with two mGRE interfaces pointing to each DMVPN cloud for high availability and redundancy.

Cisco IOS Release 12.3(13)T and Cisco IOS Release 12.3(11)T3, or later, allow multiple mGRE interfaces on a single router to be configured without tunnel keys. Each mGRE interface must reference a unique IP address as its tunnel source.

Next Hop Resolution Protocol

NHRP, defined in RFC 2332, is a Layer 2 address resolution protocol and cache, like Address Resolution Protocol (ARP) and Frame Relay Inverse ARP. NHRP is used by a branch router connected to a nonbroadcast multiaccess (NBMA) subnetwork to determine the IP address of the NBMA next hop (in this case, the hub router or the destination IP address of another branch router).

When a branch router is first established onto a DMVPN network, it registers its IP address with the hub router whose IP address is already preconfigured on the branch router. This registration enables the mGRE interface on the hub router to build a dynamic tunnel back to the registering branch router without having to know the branch tunnel destination through a command-line interface (CLI) configuration. NHRP maps a tunnel IP address to an NBMA IP address. NHRP tells the mGRE interface where to tunnel a packet to reach a certain address. When the packet is encapsulated in the mGRE packet, the IP destination address is the NBMA address.

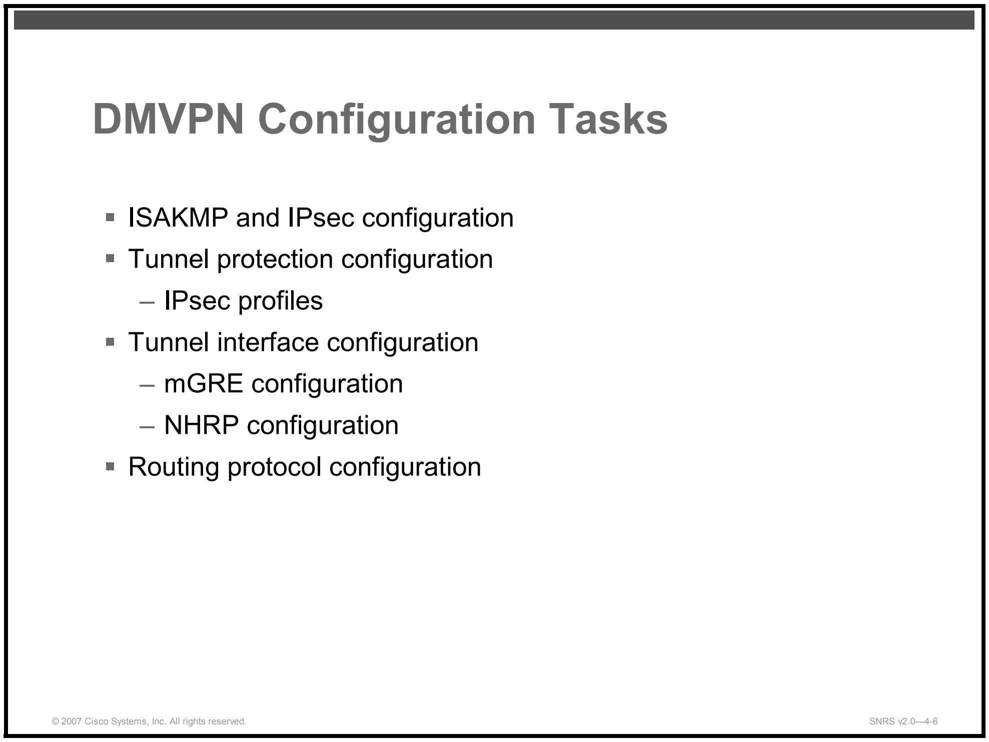
If the destination address is connected to the NBMA subnetwork, the hub router is the destination itself. Otherwise, the hub router is the egress router closest to the branch requesting a destination IP address.

Hub and branch routers should be configured with an NHRP hold time, which sets the length of time that routers instruct other routers to keep their NHRP information. This information is kept in the NHRP cache until the NHRP hold time expires and the information must be relearned. The default NHRP hold time is two hours; however, the recommended value is 10 minutes. The NHRP cache can be populated with either static or dynamic entries. On the hub router, all entries are added dynamically via registration or resolution requests. The branch router is configured with a static NHRP map pointing to the hub router. To participate in one NHRP registration process, all routers must belong to the same NHRP network by a network ID. The NHRP network ID defines an NHRP domain.

Branch routers must be configured with the NBMA address of the hub router as their Next Hop Server (NHS) to register with the hub router. The branch routers send a registration to the hub router that contains the tunnel IP address and the NBMA address. The hub router creates an entry in its NHRP cache and returns a registration reply. The branch router now views the hub router as a valid NHS and uses it as a source to locate any other branches and networks in the NHRP domain.

DMVPN Configuration Tasks

This topic describes the tasks required to configure a DMVPN.



DMVPN Configuration Tasks

- ISAKMP and IPsec configuration
- Tunnel protection configuration
  - IPsec profiles
- Tunnel interface configuration
  - mGRE configuration
  - NHRP configuration
- Routing protocol configuration

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-6

There are several tasks required when implementing a DMVPN. Listed here are the configuration tasks required to implement a DMVPN:

- Configure ISAKMP and IPsec transform sets: There must be at least one matching ISAKMP policy between two potential crypto peers. There is a default ISAKMP policy that contains the default values for the encryption algorithm, hash method (Hashed Message Authentication Code [HMAC]), Diffie-Hellman (DH) group, authentication type, and ISAKMP security association (SA) lifetime parameters. This is the lowest priority ISAKMP policy. When using pre-shared keys, Cisco recommends that wildcard keys should not be used. However, when implementing a DMVPN design using an IP address obtained dynamically, the use of a wildcard pre-shared key is required. Another approach is the use of public key infrastructure (PKI).

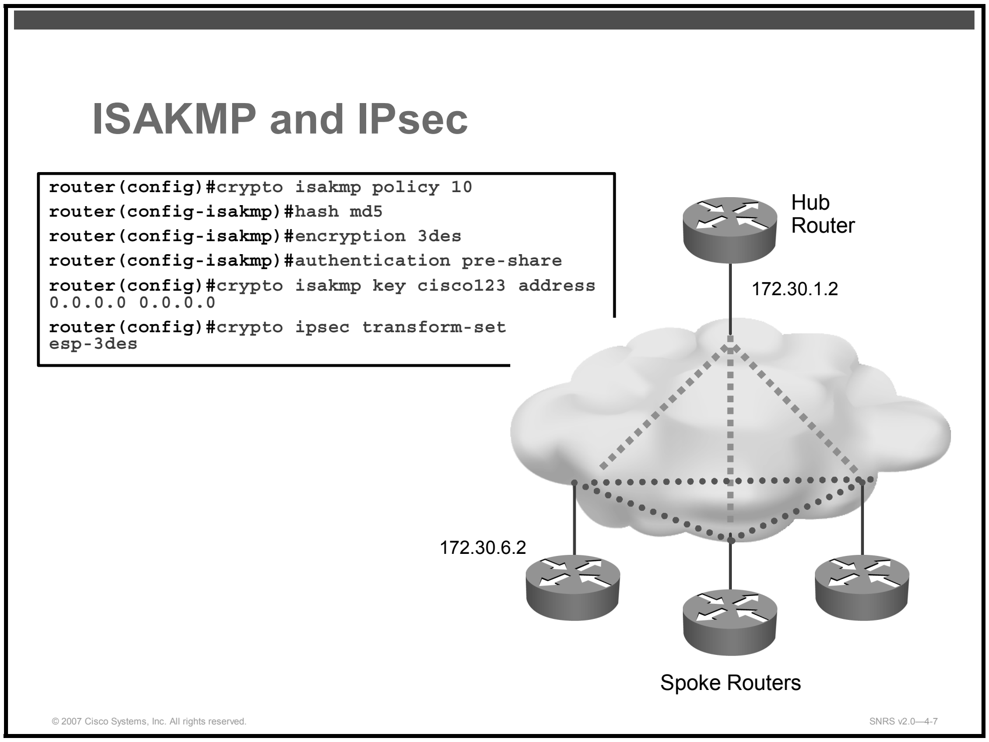
  The transform set must match between the two IPsec peers. The transform set names are locally significant only. However, the encryption algorithm, hash method, and the particular protocols used (Encapsulating Security Payload [ESP] or Authentication Header [AH]) must have at least one match. Data compression may also be configured, but it is not recommended on peers with high-speed links. There can be multiple transform sets for use between different peers, with the strongest match being negotiated.

- **Configure an IPsec profile:** IPsec profiles are used when configuring tunnel protection mode. Tunnel protection can be used when the GRE tunnel and the crypto tunnel share the same endpoints. The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands is needed in an IPsec profile. These commands pertain to an IPsec policy that can be issued under an IPsec profile; there is no need to specify the IPsec peer address or the ACL to match the packets that are to be encrypted.
- **Configure the tunnel interface:** Tunnel interface configuration includes the following tasks:
  - **Configure mGRE:** The configuration of mGRE allows a tunnel to have multiple destinations. The configuration of mGRE on one side of a tunnel does not have any relation to the tunnel properties that might exist at the exit points. This means that an mGRE tunnel on the hub may connect to a point-to-point tunnel on the branch. Conversely, a point-to-point GRE tunnel may connect to an mGRE tunnel. The distinguishing feature between an mGRE interface and a point-to-point GRE interface is the tunnel destination. An mGRE interface does not have a configured destination. Instead, the GRE tunnel is configured with the command **tunnel mode gre multipoint**. This command is used instead of the **tunnel destination** *x.x.x.x* command found with point-to-point GRE tunnels. Besides allowing for multiple destinations, an mGRE tunnel requires NHRP to resolve the tunnel endpoints.
  - **Configure NHRP:** NHRP provides a mapping between the inside and outside address of a tunnel endpoint. These mappings can be static or dynamic. In a dynamic scenario, an NHS is used to maintain a list of possible tunnel endpoints. Each endpoint using the NHS registers its own public and private mapping with the NHS. The local mapping of the NHS must always be static. It is important to note that the branch points to the inside or protected address of the NHS server.

    The NHRP hold time is used to determine how long adjacent routers should consider the cached entry of this device to be valid. The configured value is passed to the remote spoke when the spoke-to-spoke session is initiated. The remote spoke starts a countdown timer. When this timer expires, the remote router removes the cached entry to the local router. If traffic is still flowing, the remote router must request the mapping from the NHS server again. Spoke routers may have different hold times, although this practice is not common. If two spokes are in session, and one timer expires before the other, the spoke notifies the adjacent spoke that the NHRP cache entry should be aged out. Each device also removes the spoke-to-spoke encryption session.
- **Configure routing protocols:** Because the DMVPN cloud is an NBMA network, some considerations must be made when running dynamic routing protocols. This is particularly true when implementing a spoke-to-spoke design. Many routing protocols have an IP multicast mechanism that is used to discover other participating nodes. Static multicast maps are configured on branch routers pointing to the public address of the hub. The hub router is configured with a dynamic multicast map. This allows the hub and spokes to exchange broadcast information, but does not permit spokes to hear the broadcasts from other spokes.

Configuring ISAKMP and IPsec

This topic describes how to create ISAKMP policies and IPsec transforms for use with a DMVPN.



ISAKMP and IPsec

```
router(config)#crypto isakmp policy 10
router(config-isakmp)#hash md5
router(config-isakmp)#encryption 3des
router(config-isakmp)#authentication pre-share
router(config)#crypto isakmp key cisco123 address
0.0.0.0 0.0.0.0
router(config)#crypto ipsec transform-set
esp-3des
```

SNRS v2.0—4-7

There must be at least one matching ISAKMP policy between two potential crypto peers. The sample configuration in this lesson shows a policy using pre-shared keys with Data Encryption Standard (DES) as the encryption algorithm. There is a default ISAKMP policy that contains the default values for the encryption algorithm, hash method (HMAC), Diffie-Hellman group, authentication type, and ISAKMP SA lifetime parameters.

ISAKMP policies and IPsec transform sets are configured as before with any IPsec VPN. One difference though is that you can use a group (wildcard) pre-shared key to authenticate all routers using the same ISAKMP key. This is usually not advisable; however, when implementing a DMVPN design using an IP address obtained dynamically, the use of a wildcard pre-shared key is required.

This discussion includes the steps to configure ISAKMP and IPsec transforms sets using pre-shared keys and static IP addresses on the branch and hub routers.

On the hub router, follow these steps:

Step 1 Configure an IP address on the physical interface that will be the tunnel endpoint of the hub router.

```
router_hub(config)# interface FastEthernet1/0
router_hub(config-if)# ip address <public-ip-address> <mask>
```

Step 2 Create an ISAKMP policy.

```
router_hub(config)# crypto isakmp policy <priority>
```

Step 3 Specify pre-shared keys for authentication.

```
router_hub(config-isakmp)# authentication pre-share
```

Step 4 (Optional) Specify the encryption method.

```
router_hub(config-isakmp)# encryption <method>
```

Step 5 (Optional) Specify the hash algorithm.

```
router_hub(config-isakmp)# hash <algorithm>
```

Step 6 Configure ISAKMP pre-shared keys.

- To specify branch addresses individually, use this command:

```
router_hub(config)# crypto isakmp key <secret> address <spoke-
ip-address>
```

- To accept any address (wildcard pre-shared key), use this command:

```
router_hub(config)# crypto isakmp key <secret> address 0.0.0.0
```

Note When implementing a branch with a dynamic public IP address, a wildcard pre-shared key or PKI must be used on the hub router.

Step 7 Create an IPsec transform set.

```
router_hub(config)# crypto ipsec transform-set <name>
<transform>, <transform2>...
```

Configuration of the spoke routers is identical except that you will specify the hub router IP address when configuring your pre-shared key. On the spoke routers, follow these steps:

Step 1 Configure an IP address on the physical interface that will be the tunnel endpoint of the spoke router.

```
router_spoke(config)# interface FastEthernet1/0
router_spoke(config-if)# ip address <public-ip-address> <mask>
```

Step 2 Create an ISAKMP policy.

```
router_spoke(config)# crypto isakmp policy <priority>
```

Step 3 Specify pre-shared keys for authentication.

```
router_spoke(config-isakmp)# authentication pre-share
```

Step 4 (Optional) Specify the encryption method.

```
router_spoke(config-isakmp)# encryption <method>
```

Step 5 (Optional) Specify the hash algorithm.

```
router_spoke(config-isakmp)# hash <algorithm>
```

Step 6 Configure ISAKMP pre-shared keys.

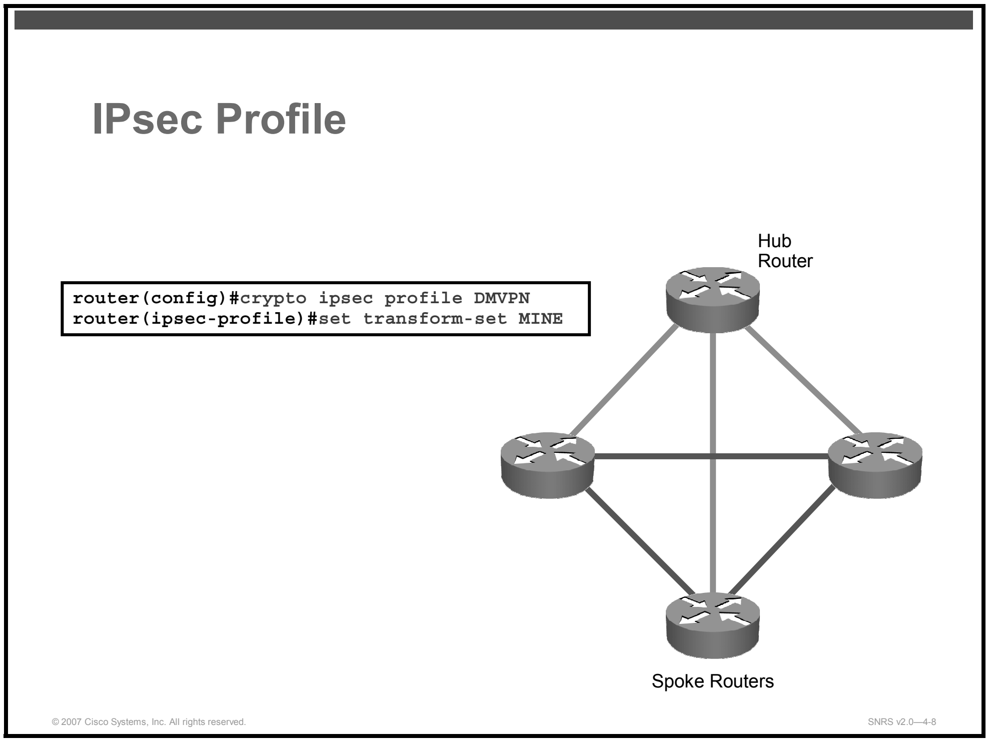
```
router_spoke(config)# crypto isakmp key <secret> address hub-
ip-address
```

Step 7 Create an IPsec transform set.

```
router_spoke(config)# crypto ipsec transform-set <name>
<transform>, <transform2>...
```

IPsec Profiles

This topic describes how to create an IPsec profile to use with tunnel protection mode.



IPsec Profile

```
router(config)#crypto ipsec profile DMVPN
router(ipsec-profile)#set transform-set MINE
```

SNRS v2.0—4-8

In typical IPsec configurations, dynamic or static crypto maps are configured on the hub and branch routers. These crypto maps specify which IPsec transform set is used and specify a crypto ACL that defines interesting traffic for the crypto map. In Cisco IOS Release 12.2(13)T or later, IPsec profiles are introduced, which share most of the same commands with the crypto map configuration; however, only a subset of the commands is needed in an IPsec profile. Only commands that pertain to an IPsec policy can be used under an IPsec profile. There is no need to specify the IPsec peer address or the ACL to match the packets that are to be encrypted.

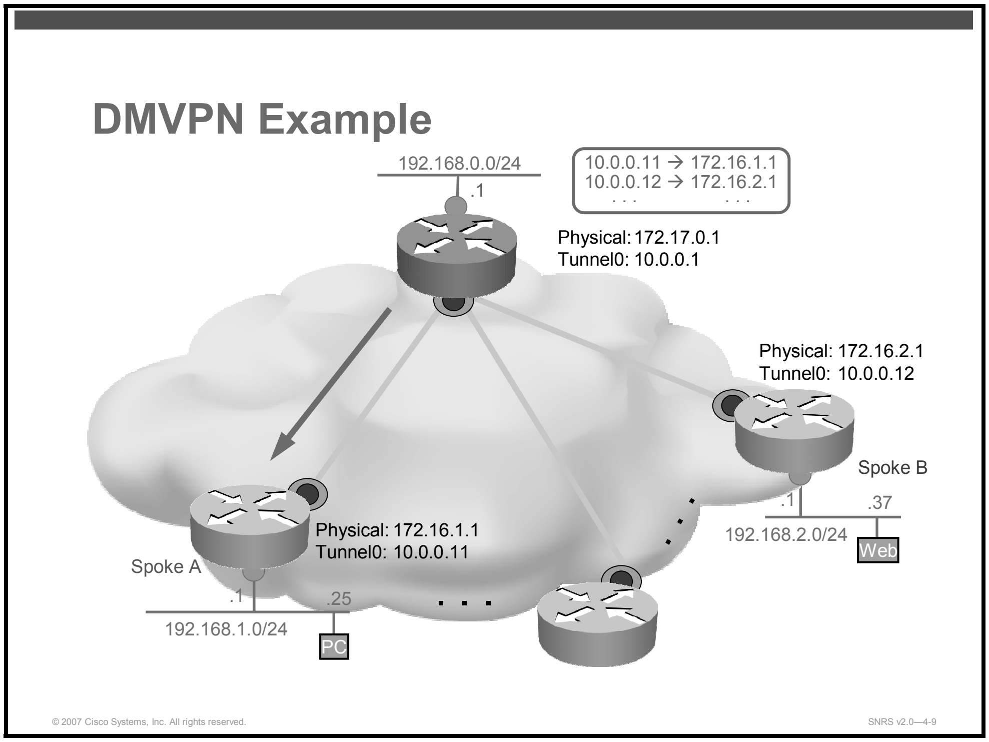
An IPsec profile will be configured on the hub and on all spoke routers. IPsec profiles abstract IPsec policy information into a single configuration entity, which can be referenced by name from other parts of the configuration. Therefore, users can configure services or features such as GRE tunnel protection with a single line of configuration. By referencing an IPsec profile, the user does not have to configure an entire crypto map configuration.

Note	An IPsec profile contains only IPsec information; that is, it does not contain any ACL information or peering information.

To enable mGRE and IPsec tunneling for hub and spoke routers, you must configure a crypto map that uses a global IPsec policy template and configure your mGRE tunnel for IPsec encryption.

DMVPN Example Operation

This section describes the DMVPN operation.



DMVPN Example

SNRS v2.0—4-9

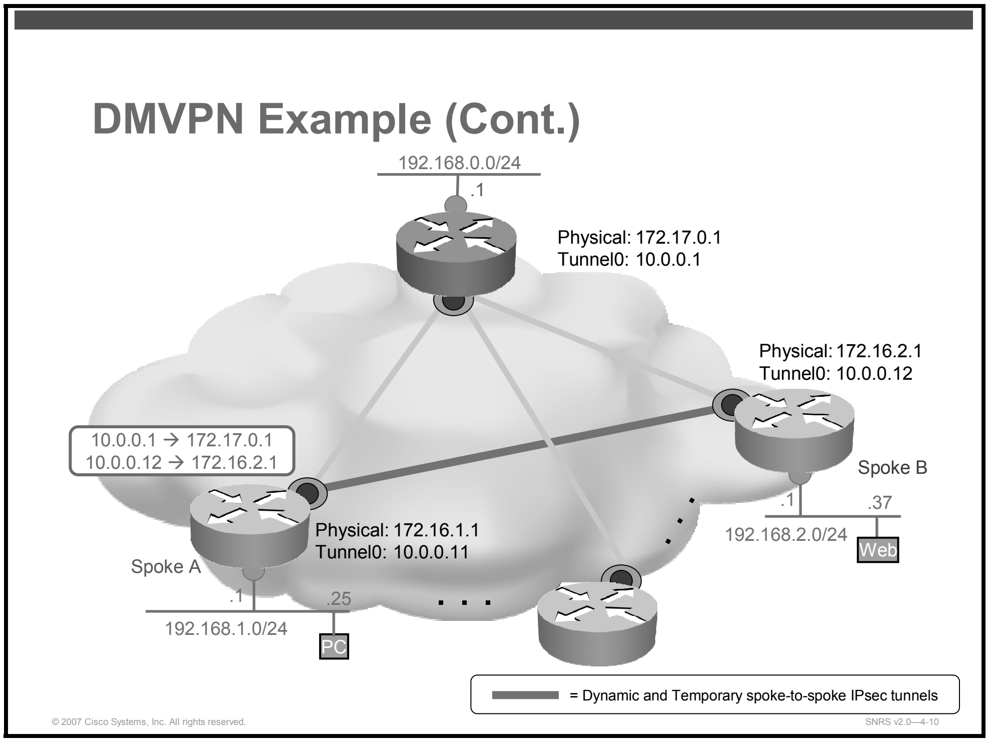
In the figure, the DMVPN example illustrates the following:

1. A PC (192.168.1.25) on the spoke A subnet wants to contact the web server (192.168.2.37) behind spoke B. It sends a packet toward the server.

2. The spoke A router consults its routing table for a route to the destination network (192.168.2.0) behind spoke B. The table provides an IP next hop of 10.0.0.12 via the tunnel0 interface of spoke A.

3. Spoke A consults its NHRP mapping table for destination 10.0.0.12 and does not find an entry. Therefore, it sends an NHRP query packet to the NHRP server.

4. The NHRP server at the hub resolves 10.0.0.12 to the corresponding public address (172.16.2.1). It sends this response to spoke A.

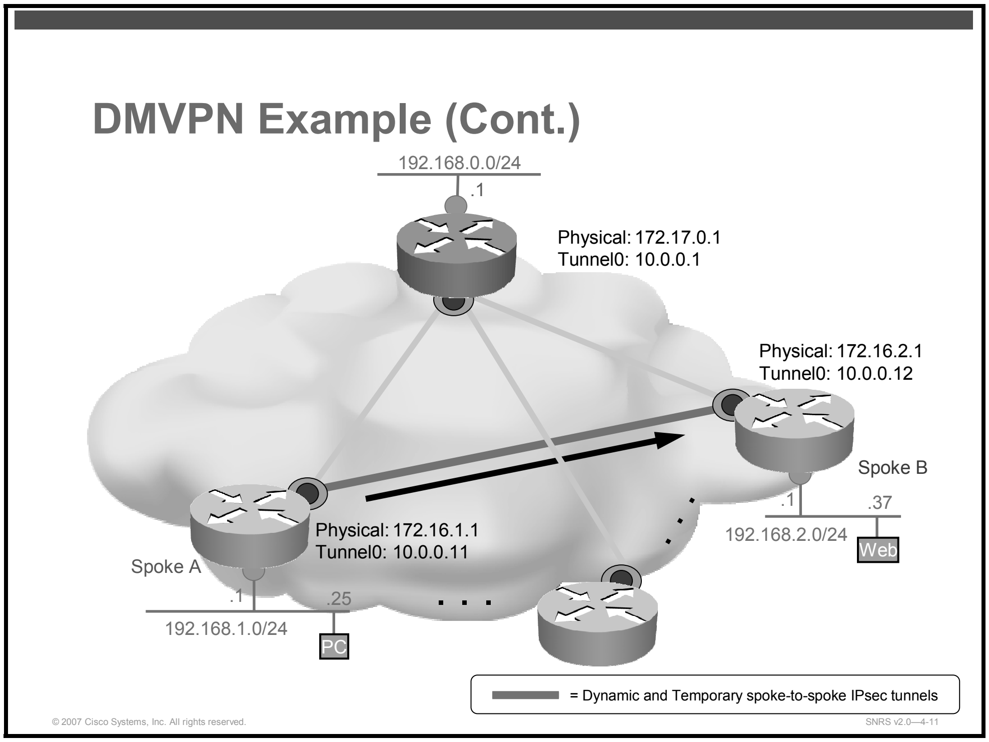


DMVPN Example (Cont.)

▬ = Dynamic and Temporary spoke-to-spoke IPsec tunnels

SNRS v2.0—4-10

5. Spoke A receives the NHRP response and enters it in its NHRP table. This triggers IPsec to create a tunnel directly to 172.16.2.1. (Spoke A uses its public address for the IPsec peer.)



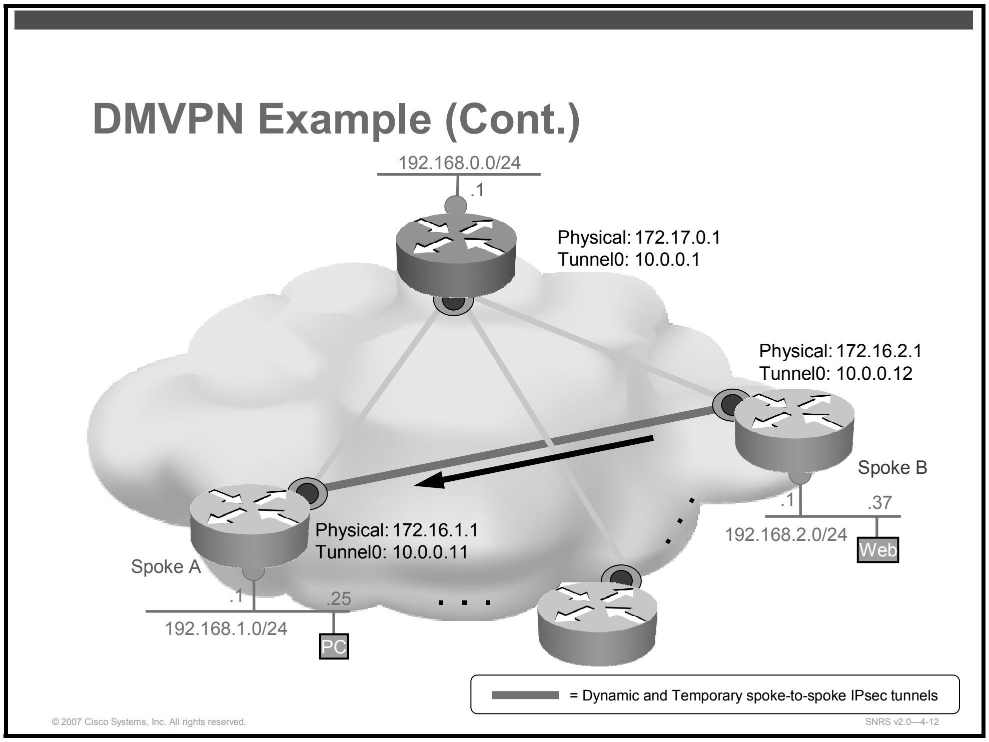
DMVPN Example (Cont.)

▬ = Dynamic and Temporary spoke-to-spoke IPsec tunnels

SNRS v2.0—4-11

6. Now that the tunnel has been built to spoke B, spoke A will send data packets to spoke B.

Note	So far, the tunnel can pass traffic in one direction only.

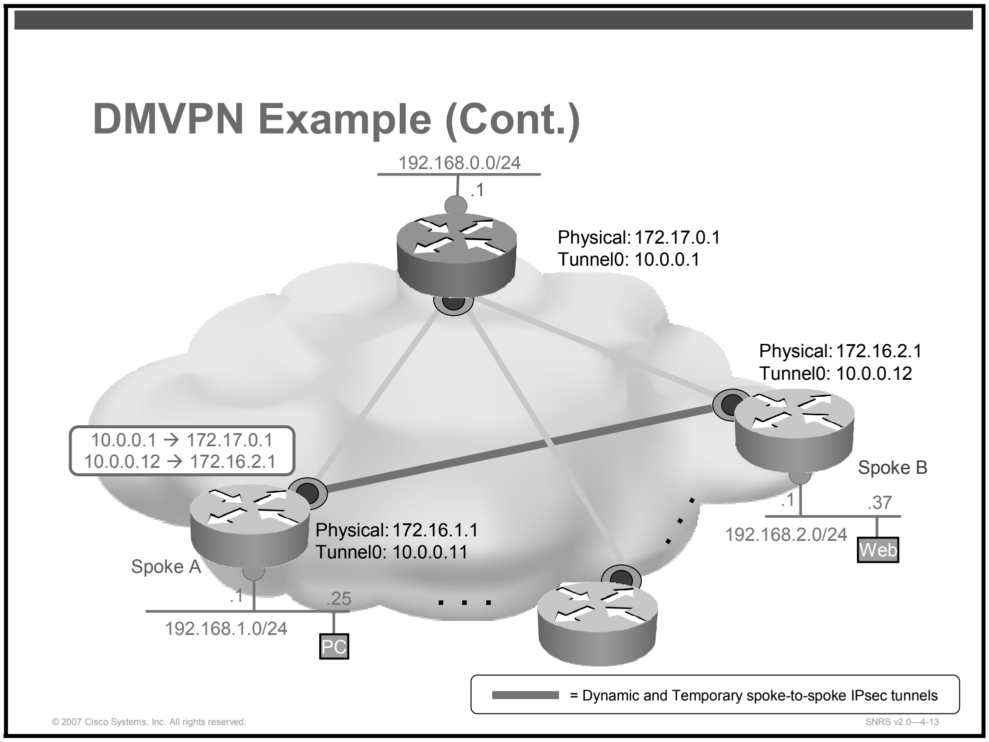


DMVPN Example (Cont.)

= Dynamic and Temporary spoke-to-spoke IPsec tunnels

SNRS v2.0—4-12

7. The web server receives the packet from the PC and sends its response. This triggers the same sequence of steps (steps 2, 3, and 4) on spoke B as was just done on spoke A. Once spoke B has the NHRP mapping for spoke A, the response packet can be sent directly to spoke A. The tunnel has already been created.

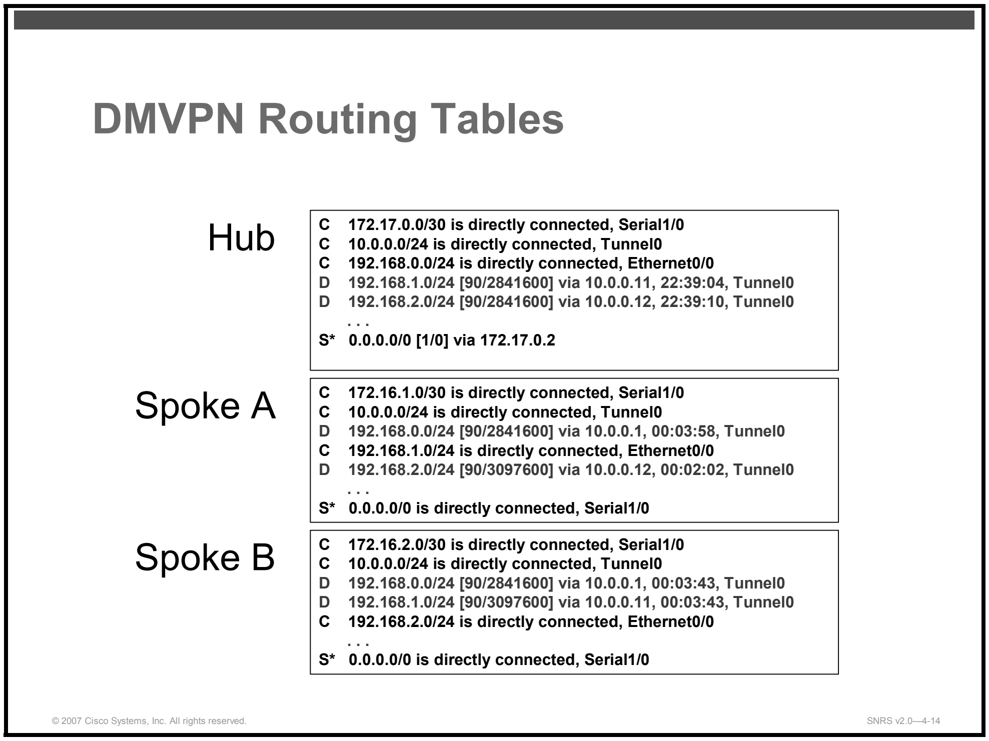


DMVPN Example (Cont.)

= Dynamic and Temporary spoke-to-spoke IPsec tunnels

SNRS v2.0—4-13

8. After a (programmable) timeout period, the NHRP entries will age out, triggering IPsec to break down the dynamic spoke-to-spoke tunnel.



DMVPN Routing Tables

Hub

```
C    172.17.0.0/30 is directly connected, Serial1/0
C    10.0.0.0/24 is directly connected, Tunnel0
C    192.168.0.0/24 is directly connected, Ethernet0/0
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, 22:39:04, Tunnel0
D    192.168.2.0/24 [90/2841600] via 10.0.0.12, 22:39:10, Tunnel0
     . . .
S*   0.0.0.0/0 [1/0] via 172.17.0.2
```

Spoke A

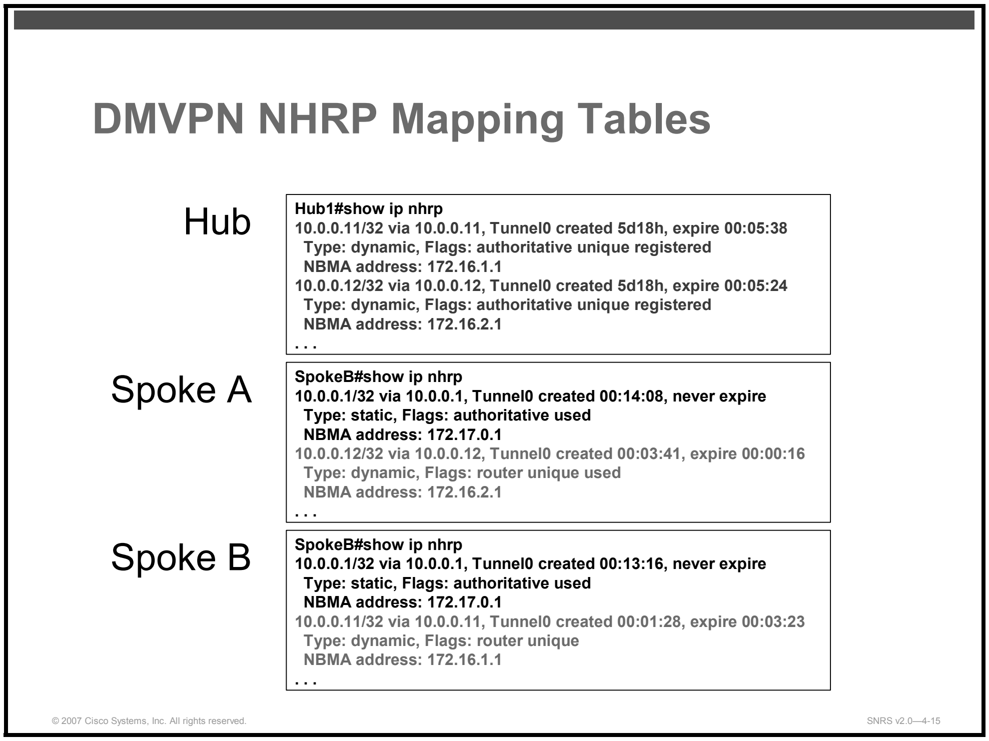
```
C    172.16.1.0/30 is directly connected, Serial1/0
C    10.0.0.0/24 is directly connected, Tunnel0
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 00:03:58, Tunnel0
C    192.168.1.0/24 is directly connected, Ethernet0/0
D    192.168.2.0/24 [90/3097600] via 10.0.0.12, 00:02:02, Tunnel0
     . . .
S*   0.0.0.0/0 is directly connected, Serial1/0
```

Spoke B

```
C    172.16.2.0/30 is directly connected, Serial1/0
C    10.0.0.0/24 is directly connected, Tunnel0
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 00:03:43, Tunnel0
D    192.168.1.0/24 [90/3097600] via 10.0.0.11, 00:03:43, Tunnel0
C    192.168.2.0/24 is directly connected, Ethernet0/0
     . . .
S*   0.0.0.0/0 is directly connected, Serial1/0
```

SNRS v2.0—4-14

These two figures show the resulting routing table and NHRP table outputs.



DMVPN NHRP Mapping Tables

Hub

```
Hub1#show ip nhrp
10.0.0.11/32 via 10.0.0.11, Tunnel0 created 5d18h, expire 00:05:38
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.16.1.1
10.0.0.12/32 via 10.0.0.12, Tunnel0 created 5d18h, expire 00:05:24
  Type: dynamic, Flags: authoritative unique registered
  NBMA address: 172.16.2.1
. . .
```

Spoke A

```
SpokeB#show ip nhrp
10.0.0.1/32 via 10.0.0.1, Tunnel0 created 00:14:08, never expire
  Type: static, Flags: authoritative used
  NBMA address: 172.17.0.1
10.0.0.12/32 via 10.0.0.12, Tunnel0 created 00:03:41, expire 00:00:16
  Type: dynamic, Flags: router unique used
  NBMA address: 172.16.2.1
. . .
```

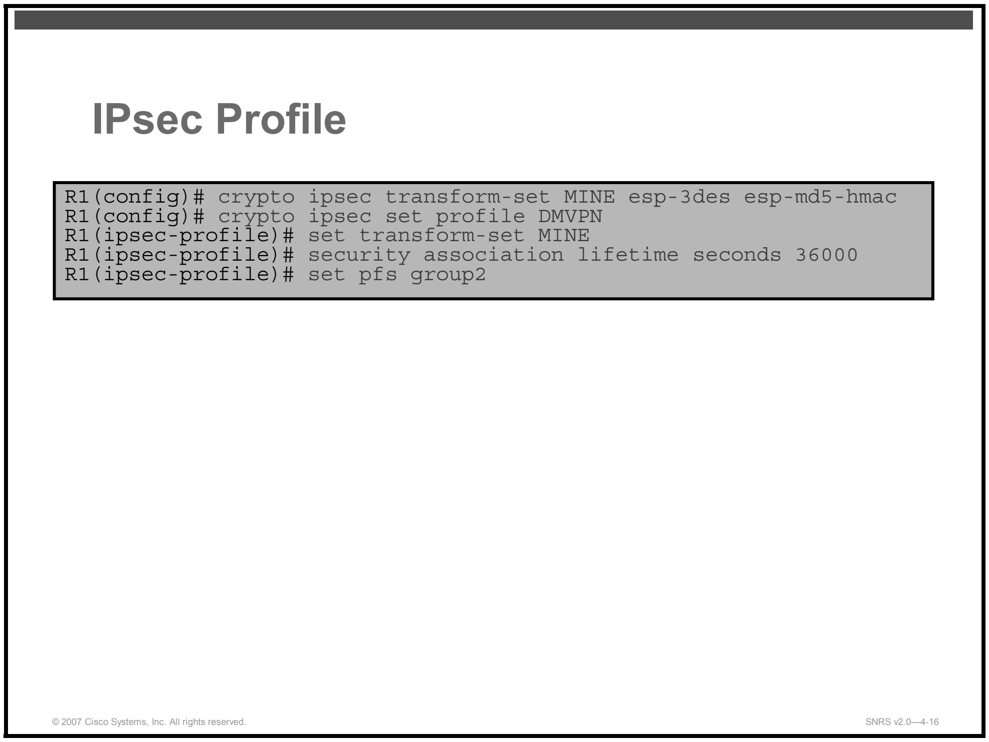
Spoke B

```
SpokeB#show ip nhrp
10.0.0.1/32 via 10.0.0.1, Tunnel0 created 00:13:16, never expire
  Type: static, Flags: authoritative used
  NBMA address: 172.17.0.1
10.0.0.11/32 via 10.0.0.11, Tunnel0 created 00:01:28, expire 00:03:23
  Type: dynamic, Flags: router unique
  NBMA address: 172.16.1.1
. . .
```

SNRS v2.0—4-15

Creating an IPsec Profile

This section describes how to create an IPsec profile to use with DMAVPN.



IPsec Profile

```
R1(config)# crypto ipsec transform-set MINE esp-3des esp-md5-hmac
R1(config)# crypto ipsec set profile DMVPN
R1(ipsec-profile)# set transform-set MINE
R1(ipsec-profile)# security association lifetime seconds 36000
R1(ipsec-profile)# set pfs group2
```

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-16

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the ACL to match the packets that are to be encrypted.

Parameters that can be configured in the IPsec profile include the following.

- **identity**: Identity restrictions.
- **isakmp-profile**: Specifies an isakmp Profile
- **pfs**: Specifies pfs settings
- **security-association:** Specifies security association parameters
- **transform-set:** Specifies a list of transform sets in priority order

Note Before configuring an IPsec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

The following steps are required to create and configure an IPsec profile:

Step 1 Create a transform set as with any other IPsec VPN configuration.

```
router(config)# crypto ipsec transform-set transform-set-name
tranform1 transform2…
```

Step 2 Create an IPsec profile and enter IPsec profile configuration mode.

```
router(config)# crypto ipsec profile name
```

Syntax Description

name	Specifies the name of the IPsec profile

Step 3 Specify which transform sets can be used with the IPsec profile.

```
router(ipsec-profile)# set transform-set transform-set-name
```

Syntax Description

transform-set-name	Specifies the name of the transform set

Step 4 (Optional) Override the global lifetime value for the IPsec profile.

```
router(ipsec-profile)# set security association lifetime
{seconds seconds | kilobytes kilobytes}
```

Syntax Description

seconds *seconds*	Specifies the number of seconds that an SA will live before expiring
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given SA before that SA expires

Step 5 (Optional) Specify that IPsec should ask for perfect forward secrecy (PFS) when requesting new SAs for this IPsec profile.

```
router(ipsec-profile)# set pfs [group1 | group2]
```

Syntax Description

group1	Specifies that IPsec should use the 768-bit DH prime modulus group when performing the new DH exchange
group2	Specifies the 1024-bit DH prime modulus group

Tunnel Protection Mode

To associate either a point-to-point GRE or mGRE tunnel with an IPsec profile on the same router, tunnel protection must be configured. Tunnel protection specifies that IPsec encryption is performed after the GRE headers are added to the tunnel packet. Tunnel protection must be configured on both the hub router and the branch router for a spoke-to-spoke deployment.

Tunnel protection can be used when the GRE tunnel and the crypto tunnel share the same endpoints.

Note GRE tunnel keepalives are not supported in combination with tunnel protection. In addition, tunnel protection cannot be used in a dual tier architecture.

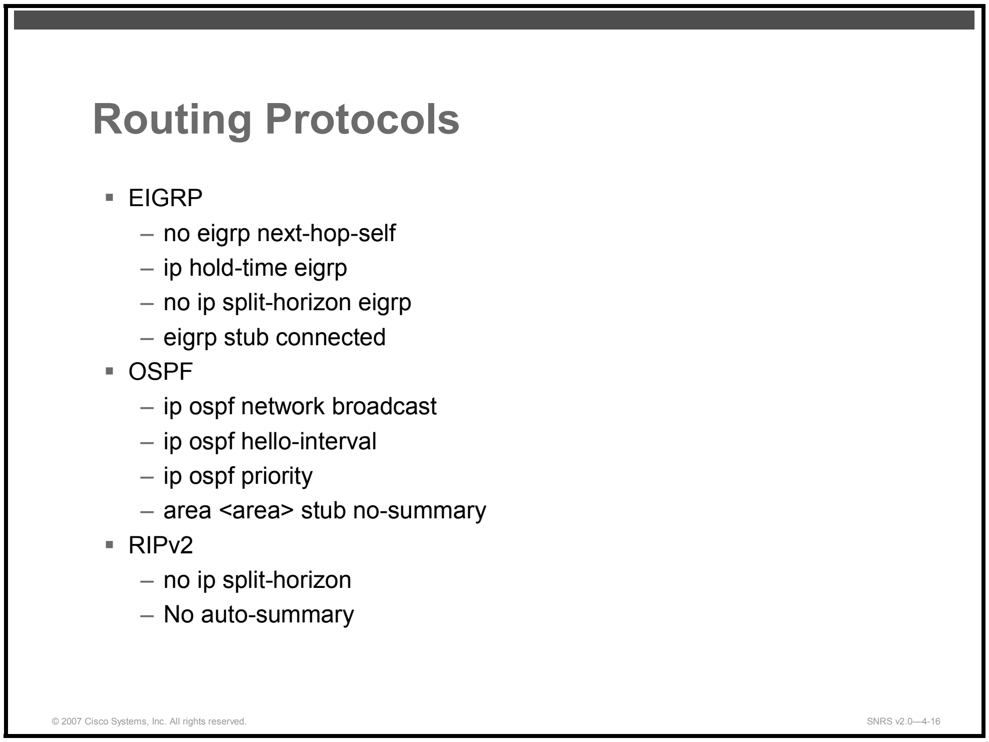
Configuring Tunnel Protection Mode

The IPsec profile is applied to the tunnel interface using the **tunnel protection ipsec profile** *profile-name* command. The **tunnel protection** command can be used with mGRE and point-to-point GRE tunnels. With point-to-point GRE tunnels, the tunnel destination address is used as the IPsec peer address. With mGRE tunnels, multiple IPsec peers are possible; the corresponding NHRP-mapped NBMA destination addresses are used as the IPsec peer addresses. Crypto ACLs that define the interesting traffic no longer need to be configured.

If more than one mGRE tunnel is configured on a router, the **shared** keyword must be configured to reference the same tunnel source address on each tunnel interface. Each mGRE tunnel interface still requires a unique tunnel key, NHRP network ID, and IP subnet address. This is common on a branch router when a dual DMVPN cloud topology is deployed.

Routing Protocols

This topic describes routing protocol issues.



Routing Protocols

- EIGRP
  - no eigrp next-hop-self
  - ip hold-time eigrp
  - no ip split-horizon eigrp
  - eigrp stub connected
- OSPF
  - ip ospf network broadcast
  - ip ospf hello-interval
  - ip ospf priority
  - area <area> stub no-summary
- RIPv2
  - no ip split-horizon
  - No auto-summary

SNRS v2.0—4-16

Some considerations must be made when running dynamic routing protocols across the DMVPN, because the DMVPN cloud is an NBMA network. This is particularly true when implementing a spoke-to-spoke design. Many routing protocols have an IP multicast mechanism that is used to discover other participating nodes. Static multicast maps are configured on branch routers pointing to the public address of the hub. The hub router is configured with a dynamic multicast map. This allows the hub and spokes to exchange broadcast information, but does not permit spokes to hear the broadcasts from other spokes.

This topic discusses DMVPN configurations using EIGRP, OSPF, and Routing Information Protocol version 2 (RIPv2).

EIGRP

EIGRP is the preferred routing protocol when running a DMVPN network. The deployment is straightforward in a pure hub-and-spoke deployment. The address space should be summarized as much as possible, and in a dual cloud topology, the spokes should be put into an EIGRP stub network. As with all EIGRP networks, the number of neighbors should be limited to ensure that the hub router can re-establish communications after a major outage. If the DMVPN subnet is configured with a /24 network prefix, the neighbor count is limited to 254, which is a safe operational limit. Beyond this number, a compromise is required to balance re-convergence with recovery. In very large EIGRP networks, it may be necessary to adjust the EIGRP hold time to allow the hub more time to recover without thrashing. However, the convergence time of the network is delayed. This method has been used in a lab to establish 400 neighbors. The maximum hold time should not exceed 7 times the EIGRP hello timer, or 35 seconds. Network designs that require the timer to be adjusted often leave little room for future growth.

Spoke-to-spoke DMVPN networks present a unique challenge because the spokes cannot directly exchange information with one another, even though they are on the same logical subnet. This limitation requires that the hub router advertise subnets from other spokes on the same subnet. This would normally be prevented by split horizon. In addition, the advertised route must contain the original next hop as learned by the hub router. A new command (**no ip next-hop-self**) was added to allow this type of operation.

Note The outside address space of the tunnel should not be included in any protocol running inside the tunnel.

Here is an example of a DMVPN configuration when EIGRP is being used as a routing protocol.

Hub

```
!
interface Tunnel0
 ip address 10.56.0.1 255.255.252.0
 ip hold-time eigrp 1 35
 no ip next-hop-self eigrp 1
 ip nhrp authentication test
 ip nhrp map multicast dynamic
 ip nhrp network-id 105600
 ip nhrp registration timeout 120
 no ip split-horizon eigrp 1
 tunnel source GigabitEthernet0/0/0
 tunnel mode gre multipoint
!
router eigrp 1
network 10.0.0.0
no auto-summary
!
```

Spokes

```
!
interface Tunnel0
 ip address 10.56.0.3 255.255.252.0
 no ip next-hop-self eigrp 1
 ip nhrp authentication test
 ip nhrp map 10.56.0.1 192.168.201.1
 ip nhrp map multicast 192.168.201.1
 ip nhrp network-id 105600
 ip nhrp nhs 10.173.20.1
 ip nhrp registration timeout 120
 no ip split-horizon eigrp 1
 tunnel source FastEthernet0/0/0
 tunnel mode gre multipoint
!
router eigrp 1
network 10.0.0.0
no auto-summary
eigrp stub connected
!
```

OSPF

Running OSPF over a DMVPN network has some of the same challenges as running OSPF over other types of networks. Historically, a single OSPF area should not contain more than 50 routers, and there should not be more than 3 areas on a router. Although current routers have stronger processors, the additional overhead of encryption and NHRP negates much of this. For this reason, the 50-router limit per area should be observed. In addition, because only the hub is in direct communications with all of the branches, it must be configured as the designated router (DR) on the DMVPN subnet. There is not typically a backup designated router (BDR). A BDR is possible if a second hub is placed on the same subnet.

The mGRE tunnel on the hub router must be configured as an OSPF broadcast network to allow the selection of a DR. Each spoke router is configured with an OSPF priority of 0 to prevent a spoke from becoming the DR. In addition, if the spoke is configured with point-to-point GRE and the hub is mGRE, the hello timer on the spoke should be changed from the default of 10 seconds to 30 seconds to match the hello timers on the mGRE interface. The tunnel IP maximum transmission unit (MTU) must match on all GRE interfaces that are OSPF-adjacent. In addition, OPSF areas running over DMVPN should be stubby areas or totally stubby areas to reduce link-state advertisement (LSA) flooding over the WAN.

In hub-and-spoke only networks, it is possible to reduce the OSPF load by using a point-to-multipoint network type on the hub router and point-to-point network type on the branch routers. In this case, there is no need to elect a DR router on the DMVPN subnet. The hub router serves as the master for the subnet. The branches consider the hub as the only path off the subnet, thus simplifying the Dijkstra's algorithm for the OPSF area.

Here is an example of a DMVPN configuration when OSPF is being used as a routing protocol

Hub

```
!
interface Tunnel0
 ip address 10.173.20.1 255.255.255.0
no ip redirects
ip mtu 1400
ip nhrp authentication secret
ip nhrp map multicast dynamic
ip nhrp network-id 10203
ip ospf network broadcast
ip ospf hello-interval 30
ip ospf priority 200
tunnel source GigabitEthernet0/1.201
tunnel mode gre multipoint
tunnel key 123
tunnel protection ipsec profile dmvpn
!
router ospf 10
network 10.173.20.0 0.0.0.255 area 10
area 10 stub no-summary
!
```

Spoke

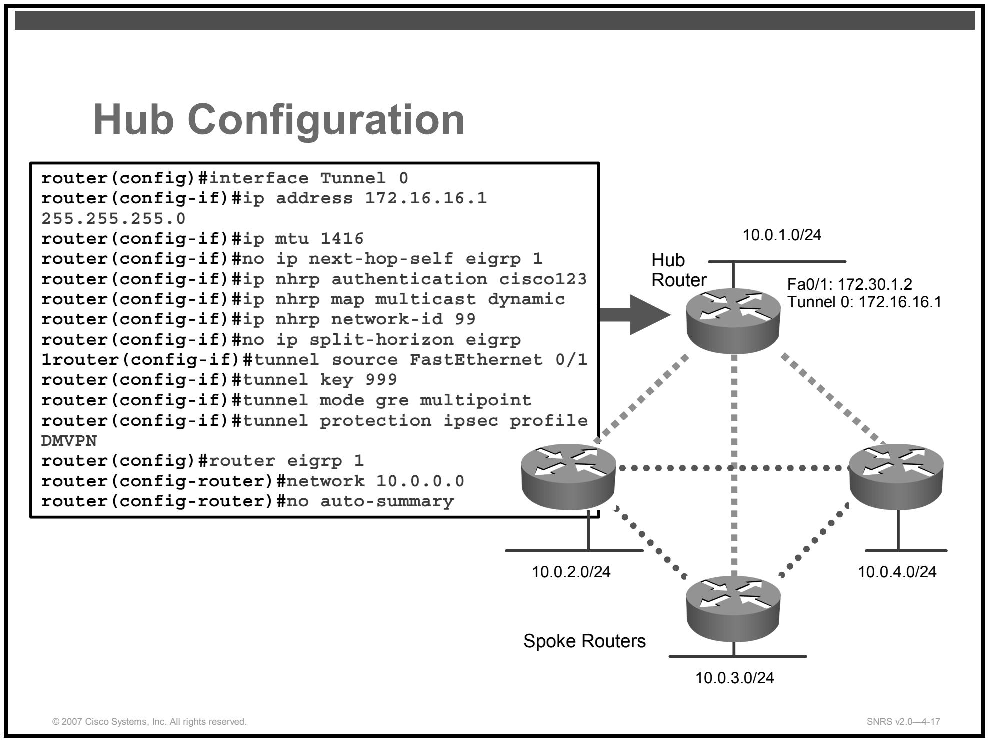
```
!
interface Tunnel0
 ip address 10.173.20.21 255.255.255.0
 ip mtu 1400
 ip nhrp authentication secret
 ip nhrp map multicast 192.168.201.1
 ip nhrp map 10.173.20.1 192.168.201.1
 ip nhrp network-id 10203
 ip nhrp nhs 10.173.20.1
 ip route-cache flow
 ip ospf network broadcast
 ip ospf hello-interval 30
 ip ospf priority 0
 qos pre-classify
 no clns route-cache
 tunnel source GigabitEthernet0/0.201
 tunnel mode gre multipoint
 tunnel key 123
 tunnel path-mtu-discovery
 tunnel protection ipsec profile dmvpn
!
router ospf 10
network 10.173.20.0 0.0.0.255 area 10
area 10 stub no-summary
!
```

RIPv2

Running RIPv2 over DMVPN is also possible. If RIPv2 is used for the routing protocol, the **no ip split-horizon** command must be configured on the hub mGRE tunnel interface if spoke-to-spoke traffic is to be permitted, even via the hub. By default, RIPv2 uses the original IP next hop instead of itself when advertising routes out the same interface from where it learned them; therefore, there is no need for a **next-hop-self** configuration. When spoke-to-spoke tunnels are in use, **auto-summary** must be disabled.

Configuring the Hub in a Spoke-to-Spoke DMVPN

This topic describes the commands used to configure the hub in a spoke-to-spoke DMVPN network.



Hub Configuration

```
router(config)#interface Tunnel 0
router(config-if)#ip address 172.16.16.1
255.255.255.0
router(config-if)#ip mtu 1416
router(config-if)#no ip next-hop-self eigrp 1
router(config-if)#ip nhrp authentication cisco123
router(config-if)#ip nhrp map multicast dynamic
router(config-if)#ip nhrp network-id 99
router(config-if)#no ip split-horizon eigrp
1router(config-if)#tunnel source FastEthernet 0/1
router(config-if)#tunnel key 999
router(config-if)#tunnel mode gre multipoint
router(config-if)#tunnel protection ipsec profile
DMVPN
router(config)#router eigrp 1
router(config-router)#network 10.0.0.0
router(config-router)#no auto-summary
```

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-17

The DMVPN hub is typically located at the company headquarters. DMVPN hub IP addresses are typically static, such as at a corporate headquarters. The spoke addresses can be static or dynamic.

To configure the hub in a DMVPN topology using pre-shared keys.

IKE and IPsec Configuration

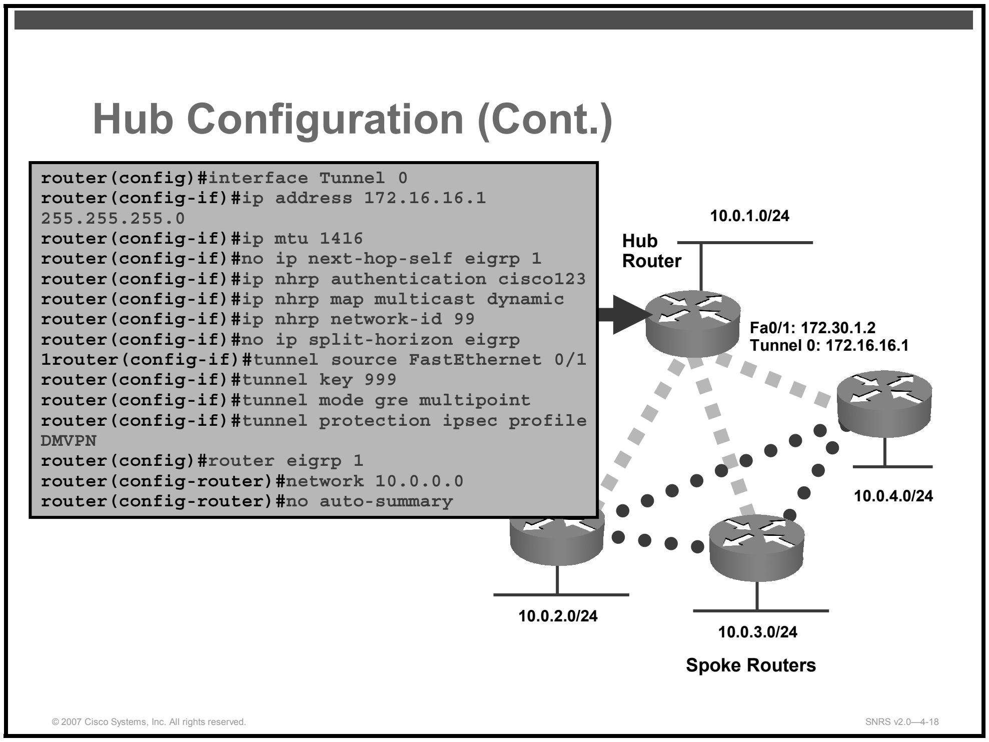
Follow these steps to configure IKE and IPsec:

Step 1 Pre-configure your ISAKMP policies as you did when setting up a site-to-site IPsec VPN using pre-shared keys. The only exception is that you will configure a group wildcard for the spokes addresses in the next step.

Step 2 Configure ISAKMP to use a group (wildcard) pre-shared key.

```
hub_router1config)# crypto isakmp key 0 key address 0.0.0.0
```

Step 3 Create an IPsec profile.



Hub Configuration (Cont.)

```
router(config)#interface Tunnel 0
router(config-if)#ip address 172.16.16.1
255.255.255.0
router(config-if)#ip mtu 1416
router(config-if)#no ip next-hop-self eigrp 1
router(config-if)#ip nhrp authentication cisco123
router(config-if)#ip nhrp map multicast dynamic
router(config-if)#ip nhrp network-id 99
router(config-if)#no ip split-horizon eigrp
1router(config-if)#tunnel source FastEthernet 0/1
router(config-if)#tunnel key 999
router(config-if)#tunnel mode gre multipoint
router(config-if)#tunnel protection ipsec profile
DMVPN
router(config)#router eigrp 1
router(config-router)#network 10.0.0.0
router(config-router)#no auto-summary
```

Tunnel Configuration

The following configuration example uses EIGRP as the routing protocol.

Step 4 Specify a tunnel interface number and enter interface configuration mode.

```
router_hub(config)# interface Tunnel 0
```

SYNTAX:

```
router(config)# interface tunnel number
```

Where:

- *number:* Specifies the number of the tunnel interface that you want to create or configure

Note There is no limit on the number of tunnel interfaces that you can create.

Step 5 Set a primary or secondary IP address for the tunnel interface.

```
router_hub(config-if)# ip address 172.16.16.1 255.255.255.0
```

SYNTAX:

```
router(config-if)# ip address ip-address mask [secondary]
```

Where:

- *secondary:* Creates a secondary address for the interface

Step 6 Set the MTU size, in bytes, of IP packets sent on the interface.

```
router_hub(config-if)# ip mtu bytes
```

Note This ensures that longer packets are fragmented before they are encrypted; otherwise, the receiving router would have to do the reassembly.

Step 7 Change the EIGRP maximum hold time so that is does not exceed 7 times the EIGRP hello timer (35 seconds).

```
router_hub(config-if)# ip hold-time eigrp 1 35
```

Step 8 Disable eigrp **next-hop-self.** The advertised route must contain the original next hop as learned by the hub router.

```
router_hub(config-if)# no ip next-hop-self eigrp 1
```

```
SYNTAX:
router(config-if)# no ip next-hop-self eigrp as-num
Where:
```

- *as-num:* Specifies the eigrp autonomous system number.

Step 9 Specify the authentication string for the interface using NHRP.

```
router_hub(config-if)# ip nhrp authentication cisco123
```

```
SYNTAX:
router(config-if)# ip nhrp authentication string
```

Step 10 Allow NHRP to automatically add spoke routers to the multicast NHRP mappings.

```
router_hub(config-if)# ip nhrp map multicast dynamic
```

```
SYNTAX:
router(config-if)# ip nhrp map multicast dynamic
```

Step 11 Enable NHRP on the interface and specify a network ID number.

```
router_hub(config-if)# ip nhrp network-id 99
```

```
SYNTAX:
router(config-if)# ip nhrp network-id number
Where:
```

- *number:* Specifies a globally unique 32-bit network identifier from an NBMA network (The range is from 1 to 4294967295.)

Step 12 Disable split horizon. This allows the hub router to advertise subnets from other spokes on the same subnet.

```
router_hub(config-if)# no ip split-horizon eigrp 1
```

Step 13 Set the source address for the tunnel interface.

```
router_hub(config-if)# tunnel source FastEthernet 0/1
```

```
SYNTAX:
router(config-if)# tunnel source {ip-address | type number}
```

Step 14 (Optional) Specify the ID key for the tunnel interface.

```
router_hub(config-if)# tunnel key 999
```

```
SYNTAX:
```

```
router(config-if)# tunnel key key-number
```

Where:

- *key-number:* Specifies a number from 0 to 4,294,967,295 that identifies the tunnel key

Note Tunnel ID keys can be used as a form of weak security to prevent improper configuration or injection of packets from a foreign source.

Step 15 Set the encapsulation mode to mGRE for the tunnel interface.

```
router_hub(config-if)# tunnel mode gre multipoint
```

SYNTAX:

```
router(config-if)# tunnel mode {gre ip | gre multipoint}
```

Where:

- **gre ip:** Specifies that GRE over IP encapsulation will be used
- **gre multipoint:** Specifies that mGRE will be used

Step 16 Associate the tunnel interface with an IPsec profile.

```
router_hub(config-if)# tunnel protection ipsec profile DMVPN
```

SYNTAX:

Step 17

```
router(config-if)# tunnel protection ipsec profile name
```

Where:

- *name:* Specifies the name of the IPsec profile (This value must match the *name* specified in the **crypto ipsec profile** *name* command.)

(If using EIGRP, you will also need to use the **no ip split-horizon eigrp *as-number*** command.)

Step 18 Enter EIGRP configuration mode.

```
router_hub(config)# router eigrp 1
```

Step 19 Specify the networks to advertise.

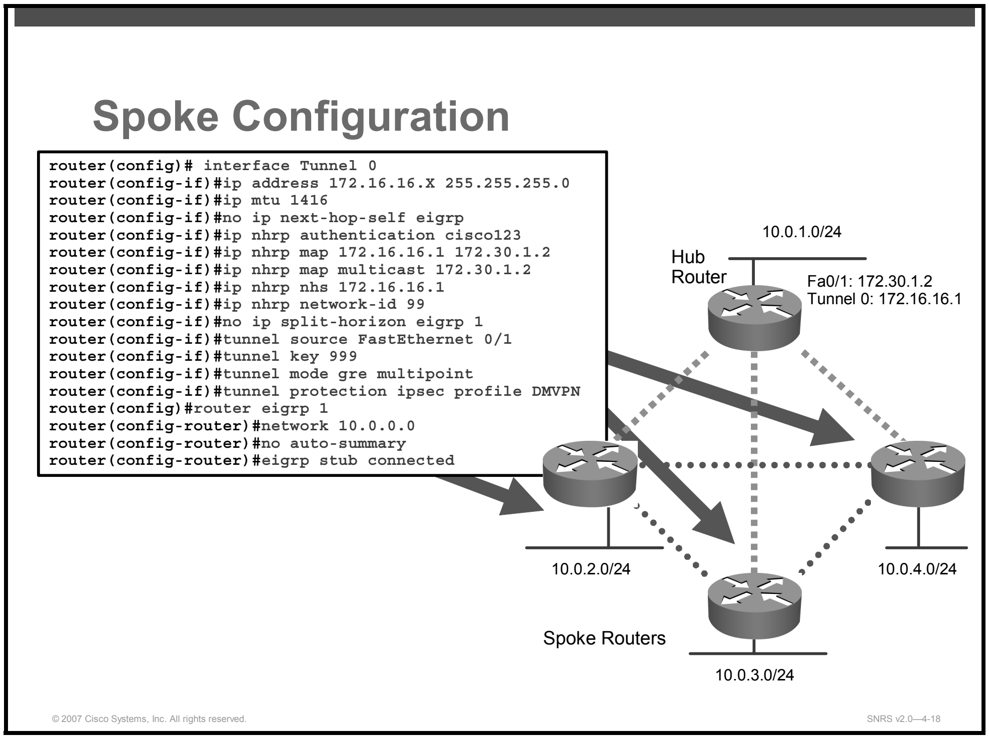
```
router_hub(config-router)# network inside-network netmask
```

Step 20 Disable auto summarization.

```
router_hub(config-router)# no auto-summary
```

Configuring a Spoke for the Spoke-to-Spoke DMVPN

This topic describes the commands used to configure the spoke in a DMVPN network.



Spoke Configuration

```
router(config)# interface Tunnel 0
router(config-if)#ip address 172.16.16.X 255.255.255.0
router(config-if)#ip mtu 1416
router(config-if)#no ip next-hop-self eigrp
router(config-if)#ip nhrp authentication cisco123
router(config-if)#ip nhrp map 172.16.16.1 172.30.1.2
router(config-if)#ip nhrp map multicast 172.30.1.2
router(config-if)#ip nhrp nhs 172.16.16.1
router(config-if)#ip nhrp network-id 99
router(config-if)#no ip split-horizon eigrp 1
router(config-if)#tunnel source FastEthernet 0/1
router(config-if)#tunnel key 999
router(config-if)#tunnel mode gre multipoint
router(config-if)#tunnel protection ipsec profile DMVPN
router(config)#router eigrp 1
router(config-router)#network 10.0.0.0
router(config-router)#no auto-summary
router(config-router)#eigrp stub connected
```

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-18

DMVPN spoke routers are typically located at branch offices of the company. Configuring the spoke routers is similar to configuring the hub except for a few different steps. When configuring the spoke routers you must do the following:

- Statically map the IP-to-NBMA address of the hub router tunnel IP address to its physical interface IP address
- Configure the spoke router to send multicast packets to the hub router
- Configure the hub router as the NHRP NHS
- Configure the authentication string for the interface using NHRP
- Configure the NHRP globally unique 32-bit network identifier

Follow these steps to configure spoke routers for mGRE and IPsec integration using pre-shared keys:

IKE and IPsec Configuration

Step 1 Pre-configure your ISAKMP and IPsec policies as you did when setting up a site-to-site IPsec VPN using pre-shared keys. You will specify the hub router address in the next step.

Step 2 Configure ISAKMP to use a pre-shared key.

```
router2(config)# crypto isakmp key 0 cisco123 address
172.30.1.2
```

Step 3 Create an IPsec profile.

Tunnel Configuration

The following configuration example uses EIGRP as the routing protocol.

Step 1 Specify a tunnel interface number and enter interface configuration mode.

```
router_spoke(config)# interface Tunnel 0
```

Step 2 Set a primary or secondary IP address for the tunnel interface.

```
router_spoke(config-if)# ip address 172.16.16.2 255.255.255.0
```

Step 3 Set the MTU size, in bytes, of IP packets sent on the interface.

```
router_spoke(config-if)# ip mtu 1416
```

Step 4 Change the EIGRP maximum hold time. This time should not exceed 7 times the EIGRP hello timer (35 seconds).

```
router_spoke(config-if)# ip mtu 1416hold-time eigrp 1 35
```

Step 5 Disable eigrp next-hop-self operation.

```
router_spoke(config-if)# no ip next-hop-self eigrp 1
```

Step 6 Configure the authentication string for the interface using NHRP.

```
router_spoke(config-if)# ip nhrp authentication cisco123
```

Step 7 Statically configure the IP-to-NBMA address mapping of the hub router.

```
router_spoke(config-if)# ip nhrp map 172.16.16.1 172.30.1.2
```

```
SYNTAX:
router(config-if)# ip nhrp map hub-tunnel-ip-address hub-physical-ip-
address
Where:
```

- *hub-tunnel-ip-address:* This is the IP address of the tunnel interface at the remote hub router.
- *hub-physical-ip-address:* This is the IP address of the physical interface at the remote hub router.

Step 8 Enable the use of a dynamic routing protocol between the spoke and hub and configure the spoke to send multicast packets to the hub router.

```
router_spoke(config-if)# ip nhrp map multicast 172.30.1.2
```

```
SYNTAX:
router(config-if)# ip nhrp map multicast hub-physical-ip-address
Where:
```

- *hub-physical-ip-address:* This is the IP address of the physical interface at the remote hub router.

Step 9 Configure the hub router as the NHRP NHS.

```
router_spoke(config-if)# ip nhrp nhs 172.16.16.1
```

SYNTAX:

```
router(config-if)# ip nhrp nhs hub-tunnel-ip-address
```

Where:

- *hub-tunnel-ip-address:* This is the IP address of the tunnel interface at the remote hub router.

Step 10 Enable NHRP on the interface and specify a network ID number.

```
router_spoke(config-if)# ip nhrp network-id 99
```

Step 11 Disable split horizon.

```
router_spoke(config-if)# no ip split-horizon eigrp 1
```

Step 12 Set the source address for the tunnel interface.

```
router_spoke(config-if)# tunnel source FastEthernet 0/1
```

Step 13 Specify the ID key for the tunnel interface.

```
router_spoke(config-if)# tunnel key 999
```

Step 14 Set the encapsulation mode to mGRE for the tunnel interface.

```
router_spoke(config-if)# tunnel mode gre multipoint
```

Step 15 Associate the tunnel interface with an IPsec profile.

```
router_spoke(config-if)# tunnel protection ipsec profile DMVPN
```

Step 16 Enter EIGRP configuration mode.

```
router_spoke(config)# router eigrp 1
```

Step 17 Specify networks to advertise.

```
Router_spoke(config-router)# network inside-network netmask
```

Step 18 Disable auto summarization.

```
router_spoke(config)# no auto-summary
```

Step 19 Configure the router as a stub. Use this command to configure a router as a stub where the router directs all IP traffic to a distribution router.

```
router_spoke(config-router)# eigrp stub connected
```

SYNTAX:

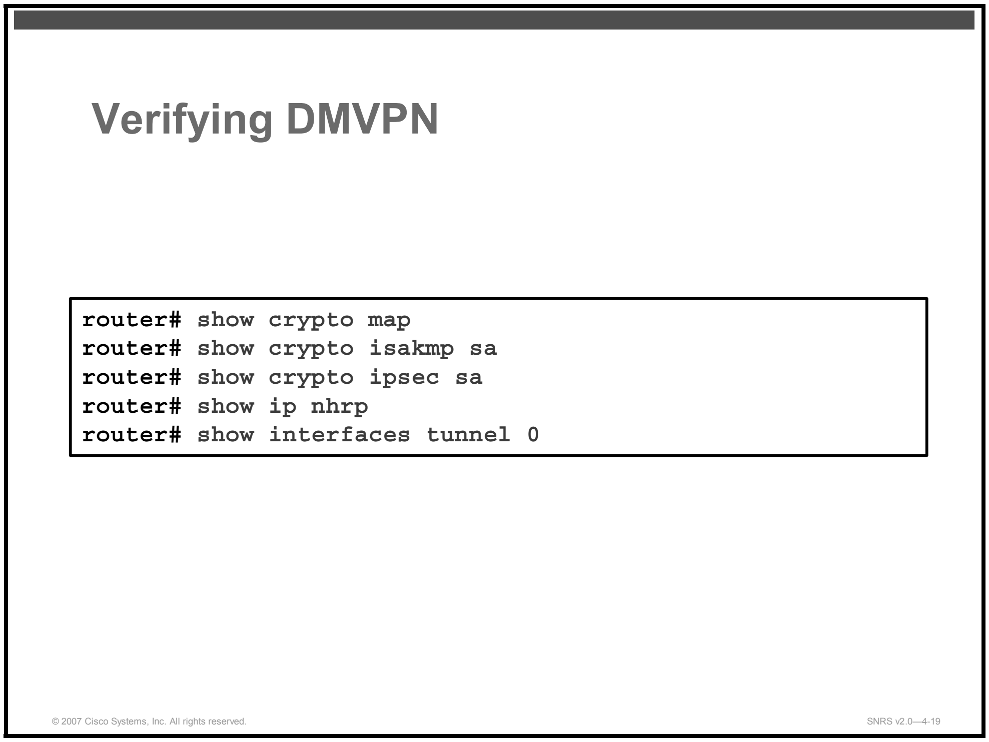
```
router(config-router)# eigrp stub [receive-only | connected | static |
summary | redistributed]
```

Where:

- **receive-only:** Sets the router as a receive-only neighbor
- **connected:** Advertises connected routes
- **static:** Advertises static routes
- **summary:** Advertises summary routes
- **redistributed:** Advertises redistributed routes from other protocols and autonomous systems

Verifying DMVPN

This topic describes how to verify DMVPN connectivity.



Verifying DMVPN

```
router# show crypto map
router# show crypto isakmp sa
router# show crypto ipsec sa
router# show ip nhrp
router# show interfaces tunnel 0
```

There are several commands available to verify and troubleshoot DMVPN operation and configuration.

The following commands are useful in verifying DMVPM operation:

- **show crypto map**
- **show ip nhrp**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

Hub Verification

After the spokes have been configured, you should see the SAs that have been negotiated between the hub and the spokes. To verify the operation at the hub router, perform the following commands and observe the output:

```
router_hub# show crypto map
Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
         Profile name: DMVPN
         Security association lifetime: 4608000 kilobytes/3600 seconds
         PFS (Y/N): N
         Transform sets={
                 MINE,
         }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
         Map is a PROFILE INSTANCE.
         Peer = 172.30.1.5
         Extended IP access list
             access-list  permit gre host 172.30.1.2 host 172.30.1.5
         Current peer: 172.30.1.5
         Security association lifetime: 4608000 kilobytes/3600 seconds
         PFS (Y/N): N
         Transform sets={
                 MINE,
         }
         Interfaces using crypto map Tunnel0-head-0:
                 Tunnel0

router_hub# show ip nhrp
172.16.16.6/32 via 172.16.16.6, Tunnel0 created 01:12:15, expire
01:27:44
  Type: dynamic, Flags: unique nat registered
  NBMA address: 172.30.1.5
172.16.16.7/32 via 172.16.16.7, Tunnel0 created 00:55:34, expire
01:44:25
  Type: dynamic, Flags: unique registered
  NBMA address: 172.30.1.6

router_hub# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.1.2      172.30.1.5      QM_IDLE           1002    0 ACTIVE

IPv6 Crypto ISAKMP SA

router_hub# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.2
   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
```

```
   remote ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)

   current_peer 172.30.1.5 port 500

     PERMIT, flags={origin_is_acl,}

    #pkts encaps: 1, #pkts encrypt: 1, #pkts digest: 1

    #pkts decaps: 1, #pkts decrypt: 1, #pkts verify: 1

    #pkts compressed: 0, #pkts decompressed: 0

    #pkts not compressed: 0, #pkts compr. failed: 0

    #pkts not decompressed: 0, #pkts decompress failed: 0

    #send errors 0, #recv errors 0

     local crypto endpt.: 172.30.1.2, remote crypto endpt.: 172.30.1.5

     path mtu 1500, ip mtu 1500

     current outbound spi: 0xDE42FC43(3728931907)


     inbound esp sas:

      spi: 0x3BC2CC59(1002622041)

        transform: esp-des ,

        in use settings ={Tunnel, }

        conn id: 2003, flow_id: FPGA:3, crypto map: Tunnel0-head-0

        sa timing: remaining key lifetime (k/sec): (4543748/2956)

        IV size: 8 bytes

        replay detection support: N

        Status: ACTIVE

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

      spi: 0xDE42FC43(3728931907)

        transform: esp-des ,

        in use settings ={Tunnel, }

        conn id: 2004, flow_id: FPGA:4, crypto map: Tunnel0-head-0

        sa timing: remaining key lifetime (k/sec): (4543748/2944)

        IV size: 8 bytes

        replay detection support: N

        Status: ACTIVE

     outbound ah sas:

     outbound pcp sas:

router_hub# show ip nhrp

172.16.16.2/32 via 172.16.16.2, Tunnel0 created 00:11:51, expire
01:48:08

  Type: dynamic, Flags: unique nat registered
  NBMA address: 172.30.1.5
```

You may use the same commands for spoke router verification.

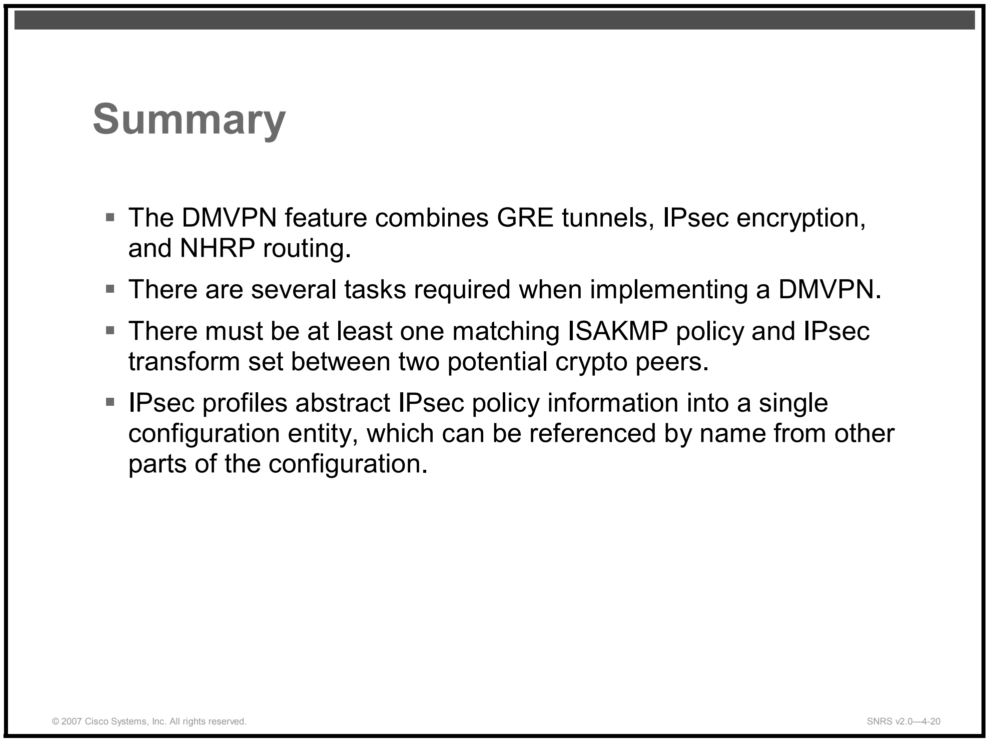
Spoke Verification

To verify the operation at the spoke routers, perform the following commands and observe the output:

```
spoke# show crypto map
Crypto Map "MYMAP" 10 ipsec-isakmp
        Peer = 172.30.1.2
        Extended IP access list vpn
            access-list vpn permit ip host 172.30.1.5 host 172.30.1.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map MYMAP:

Crypto Map "Tunnel0-head-0" 65536 ipsec-isakmp
        Profile name: DMVPN
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }

Crypto Map "Tunnel0-head-0" 65537 ipsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.30.1.2
        Extended IP access list
            access-list  permit gre host 172.30.1.5 host 172.30.1.2
        Current peer: 172.30.1.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                MINE,
        }
        Interfaces using crypto map Tunnel0-head-0:
                Tunnel0

router_spoke# show ip nhrp
172.16.16.1/32 via 172.16.16.1, Tunnel0 created 01:32:20, never expire
  Type: static, Flags: nat used
  NBMA address: 172.30.1.
172.16.16.6/32 via 172.16.16.6, Tunnel0 created 00:06:52, expire
01:53:07
  Type: dynamic, Flags: router unique nat local
  NBMA address: 172.30.1.5
    (no-socket)
172.16.16.7/32 via 172.16.16.7, Tunnel0 created 00:06:53, expire
01:53:07
  Type: dynamic, Flags: router implicit
  NBMA address: 172.30.1.6

router_spoke# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
172.30.1.2      172.30.1.5      QM_IDLE           1002    0 ACTIVE

IPv6 Crypto ISAKMP SA

router2_spoke# show crypto ipsec sa
interface: Tunnel0
    Crypto map tag: Tunnel0-head-0, local addr 172.30.1.5
   protected vrf: (none)
   local  ident (addr/mask/prot/port):
(172.30.1.5/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port):
(172.30.1.2/255.255.255.255/47/0)
   current_peer 172.30.1.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 1, #pkts encrypt: 1, #pkts digest: 1
    #pkts decaps: 1, #pkts decrypt: 1, #pkts verify: 1
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
     local crypto endpt.: 172.30.1.5, remote crypto endpt.: 172.30.1.2
     path mtu 1500, ip mtu 1500
     current outbound spi: 0x3BC2CC59(1002622041)
     inbound esp sas:
      spi: 0xDE42FC43(3728931907)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2003, flow_id: FPGA:3, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4522082/3415)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0x3BC2CC59(1002622041)
        transform: esp-des ,
        in use settings ={Tunnel, }
        conn id: 2004, flow_id: FPGA:4, crypto map: Tunnel0-head-0
        sa timing: remaining key lifetime (k/sec): (4522082/3407)
        IV size: 8 bytes
        replay detection support: N
        Status: ACTIVE
     outbound ah sas:
     outbound pcp sas:

router_spoke# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.16.7/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 172.30.1.6 (FastEthernet0/1), destination UNKNOWN

Show  Tunnel protocol/transport multi-GRE/IP, key 0x3E7, sequencing
disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "DMVPN")
  Last input 00:02:11, output 00:02:11, output hang never
  Last clearing of "show interface tunnel 0" counters 00:36:12
Etc.............
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7 packets input, 940 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     7 packets output, 864 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

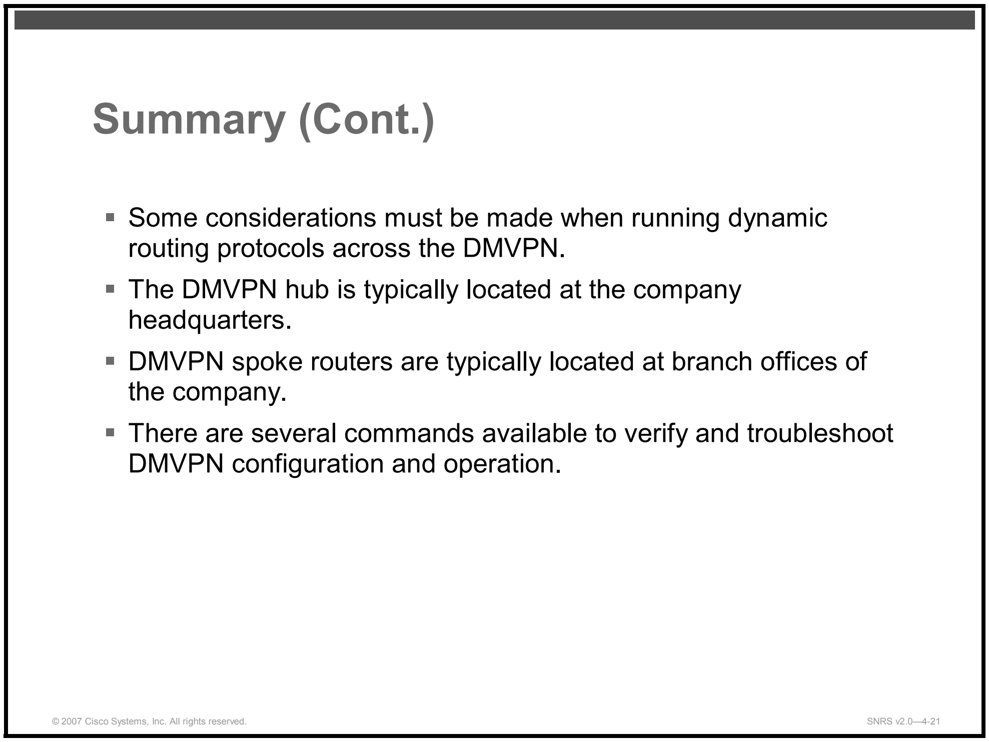
Summary

This topic summarizes the key points that were discussed in this lesson.



Summary

- The DMVPN feature combines GRE tunnels, IPsec encryption, and NHRP routing.
- There are several tasks required when implementing a DMVPN.
- There must be at least one matching ISAKMP policy and IPsec transform set between two potential crypto peers.
- IPsec profiles abstract IPsec policy information into a single configuration entity, which can be referenced by name from other parts of the configuration.



Summary (Cont.)

- Some considerations must be made when running dynamic routing protocols across the DMVPN.
- The DMVPN hub is typically located at the company headquarters.
- DMVPN spoke routers are typically located at branch offices of the company.
- There are several commands available to verify and troubleshoot DMVPN configuration and operation.

References

For additional information, refer to this resource:

- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4: Dynamic Multipoint VPN (DMVPN)* http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_chapter09186a0080455c71.html.

Lesson 7

Configuring Cisco IOS SSL VPN (WebVPN)

Overview

The Cisco IOS Software Secure Sockets Layer (SSL) Virtual Private Network (VPN) (also known as WebVPN) allows users to remotely connect to a corporate network without the use of a preloaded VPN client on their computers. In this lesson, you will learn how to configure a Cisco IOS router to use the WebVPN feature to create a corporate portal that can be accessed from anywhere on the Internet.

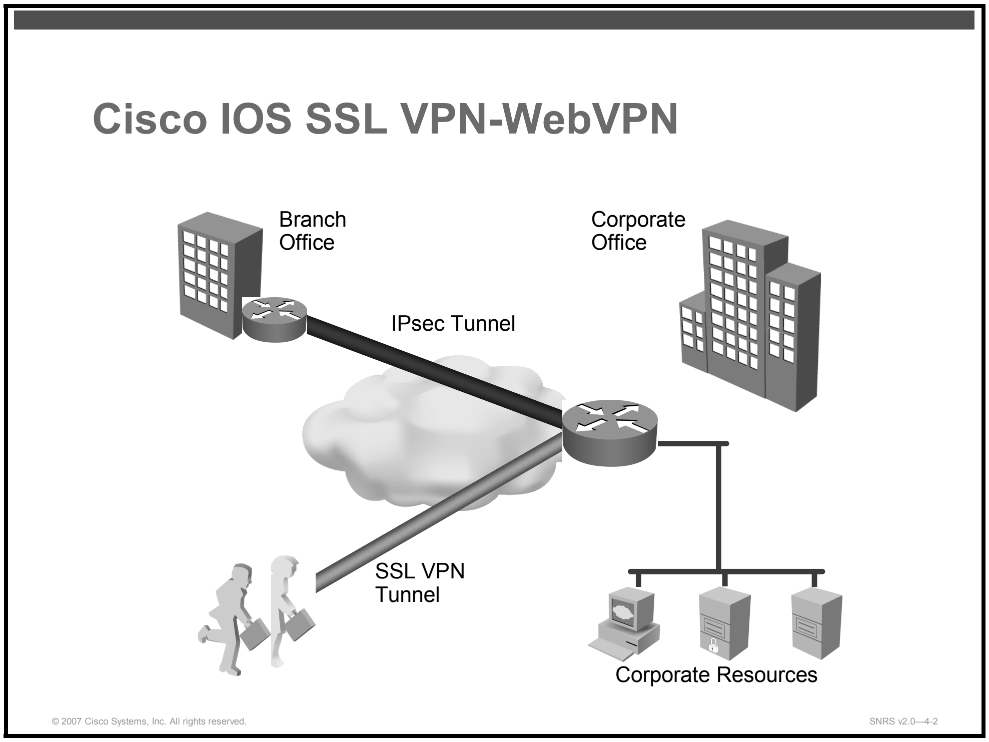
Objectives

Upon completing this lesson, you will be able to describe and configure the operation of a Cisco IOS SSL VPN. This ability includes being able to meet these objectives:

- Describe the overall WebVPN feature
- Describe the clientless access mode of WebVPN
- Describe the thin-client or port forwarding access mode of WebVPN
- Describe the full network access (tunnel) mode of WebVPN
- List the tasks required to configure WebVPN for clientless and thin-client access
- Describe how to configure AAA for WebVPN
- Describe how to configure DNS for WebVPN
- Describe the process for configuring certificates and trustpoints for WebVPN
- Describe how to configure WebVPN services on a Cisco IOS router
- Describe the commands used to monitor and maintain WebVPN
- Describe the commands used to troubleshoot WebVPN

Overview of Cisco IOS SSL VPN (WebVPN)

This topic describes the overall WebVPN feature.



Cisco IOS SSL VPN-WebVPN

SNRS v2.0—4-2

The Cisco IOS SSL VPN (WebVPN) feature, in Cisco IOS Software, provides support for remote user access to enterprise networks from anywhere on the Internet. Remote access is provided through a SSL-enabled WebVPN gateway. The WebVPN gateway allows remote users to establish a secure VPN tunnel using a web browser. This feature provides a comprehensive solution that allows easy access to a broad range of web resources and web-enabled applications using native secure HTTP (HTTPS) (HTTP over SSL) browser support. WebVPN delivers three modes of SSL VPN access: clientless, thin-client, and full-tunnel client (Cisco SSL VPN Client [SVC]) support.

Cisco IOS WebVPN can also support access from noncorporate-owned machines, including home computers, Internet kiosks, and wireless hot spots. These locations are difficult places to deploy and manage VPN client software and remote configuration required to support IP Security (IPsec) VPN connections.

Using SSL VPNs, site-to-site IPsec connectivity between the main and remote sites is unaltered while the mobile worker needs only Internet access and supported software (web browser and OS) to securely access the corporate network.

Prerequisites

To use WebVPN, the following prerequisites must be adhered to:

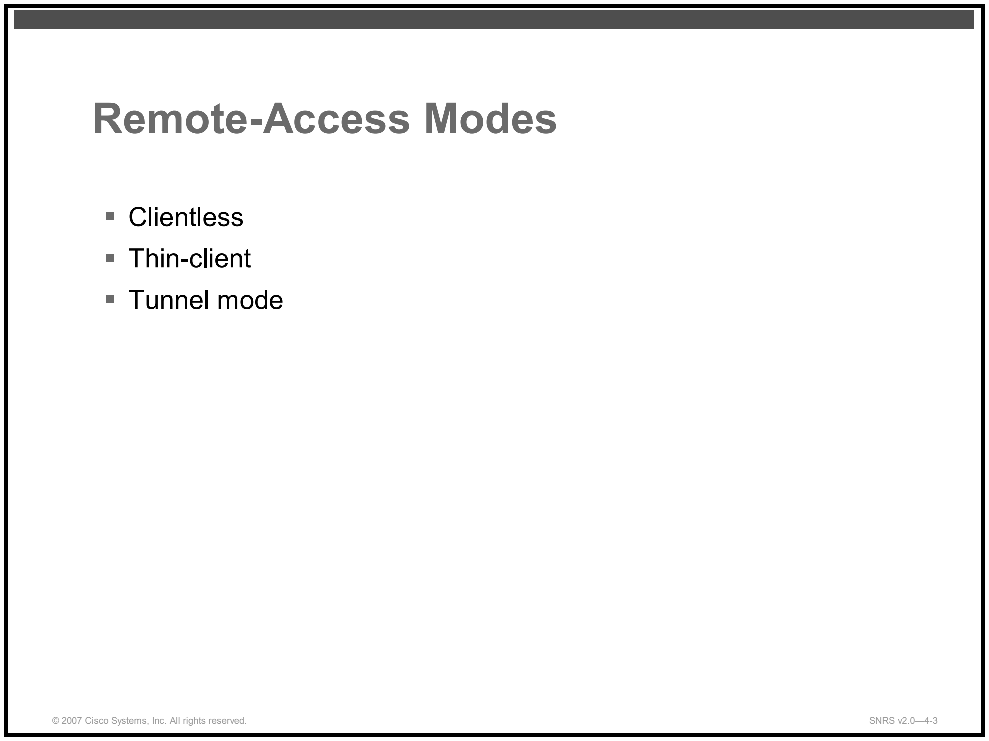
- To securely access resources on a private network behind a WebVPN gateway, the remote user of a WebVPN service must have the following:
  - An account (login name and password)
  - An SSL-enabled browser (for example, Internet Explorer, Netscape, Mozilla, or Mozilla Firefox)
  - E-mail client, such as Eudora, Microsoft Outlook, or Netscape Mail
  - The Microsoft Windows 2000 or Microsoft Windows XP OS with either the Sun Microsystems Java Runtime Environment (JRE) for Microsoft Windows version 1.4 or later or a browser that supports ActiveX control.

    Or
  - The Linux operating system with Sun MicroSystems JRE for Linux version 1.4 or later. To access Microsoft file shares from Linux in clientless remote-access mode, Samba must also be installed.
- Thin-client support used for TCP port-forwarding applications requires administrative privileges on the computer of the remote user.
- Full tunnel client support used for Cisco IOS SSL VPN access requires administrative privileges on the computer of the remote user.
- The remote user must have local administrative privileges to use thin-client or full tunnel client features.
- The WebVPN gateway and context configuration must be completed before a remote user can access resources on a private network behind a WebVPN.

Restrictions

Here are some of the restrictions for using WebVPN:

- URLs referred by the Macromedia Flash player cannot be modified for secure retrieval by the WebVPN gateway.
- If WebVPN has to be enabled on a router that is running HTTPS server, the administrator must configure an IP address for WebVPN under the **webvpn gateway** configuration mode using the **ip addr** keyword option.
- Thin client used for TCP port-forwarding applications requires administrative privileges on the computer of the end user.



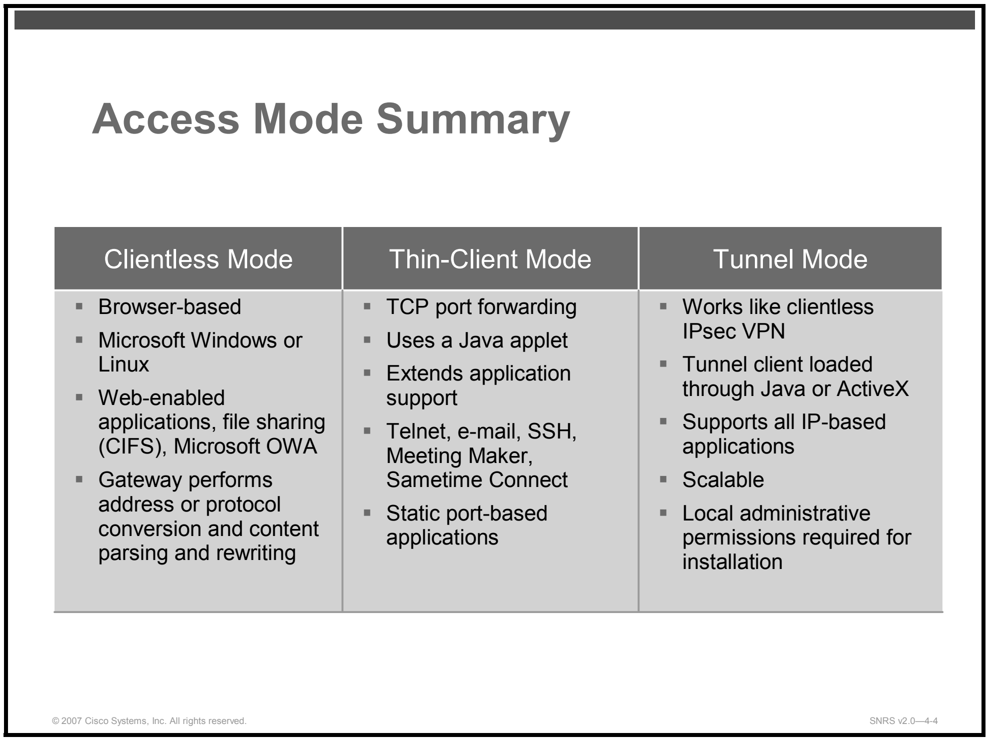
Remote-Access Modes

- Clientless
- Thin-client
- Tunnel mode

SNRS v2.0—4-3

WebVPN delivers the following three modes of Cisco IOS SSL VPN access:

- **Clientless:** Clientless mode provides secure access to private web resources and will provide access to web content. This mode is useful for accessing most of the content that you would expect to access in a web browser, such as Internet access, databases, and online tools that employ a web interface.
- **Thin-client (port-forwarding Java applet):** Thin-client mode extends the capability of the cryptographic functions of the web browser to enable remote access to TCP-based applications such as Post Office Protocol version 3 (POP3), Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP), Telnet, and Secure Shell (SSH).
- **Tunnel mode:** Full tunnel client mode offers extensive application support through its dynamically downloaded Cisco SVC for WebVPN. Full tunnel client mode delivers a lightweight, centrally configured and easy-to-support SSL VPN tunneling client that provides network layer access to virtually any application.



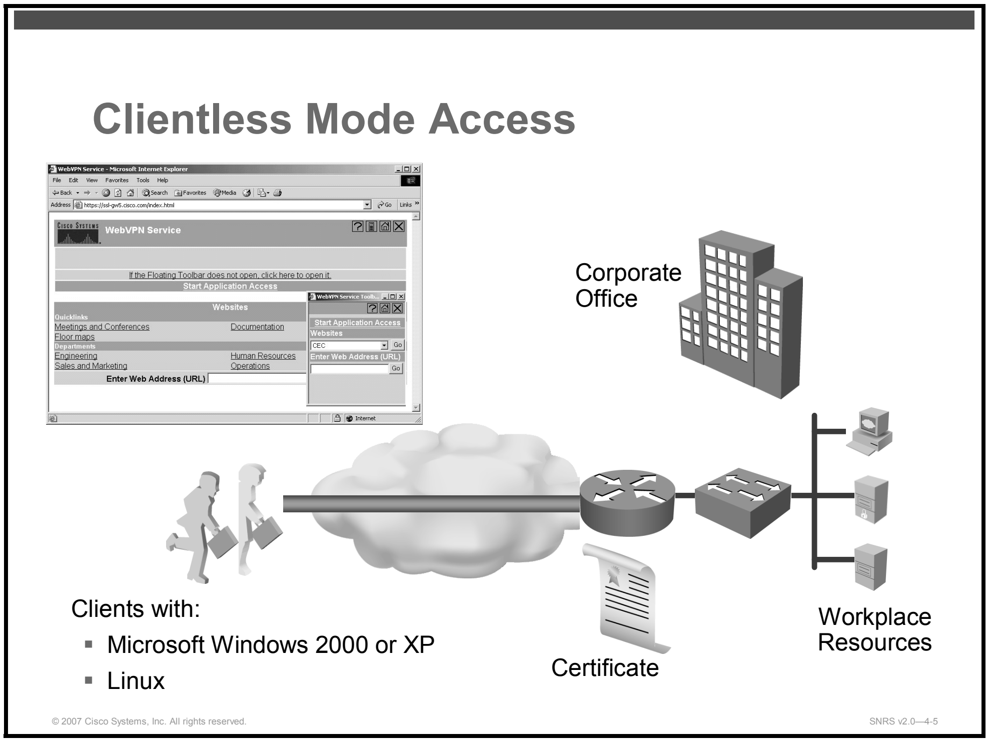
Access Mode Summary

Clientless Mode	Thin-Client Mode	Tunnel Mode
▪ Browser-based ▪ Microsoft Windows or Linux ▪ Web-enabled applications, file sharing (CIFS), Microsoft OWA ▪ Gateway performs address or protocol conversion and content parsing and rewriting	▪ TCP port forwarding ▪ Uses a Java applet ▪ Extends application support ▪ Telnet, e-mail, SSH, Meeting Maker, Sametime Connect ▪ Static port-based applications	▪ Works like clientless IPsec VPN ▪ Tunnel client loaded through Java or ActiveX ▪ Supports all IP-based applications ▪ Scalable ▪ Local administrative permissions required for installation

End-user login and authentication is performed by the web browser to the secure gateway using an HTTP request. This process creates a session that is referenced by a cookie. After authentication, the remote user is shown a portal page that allows access to the WebVPN networks. All requests sent by the browser include the authentication cookie. The portal page provides all of the resources available on the internal networks. For example, the portal page could provide a link to allow the remote user to download and install a thin-client Java applet (for TCP port forwarding) or a tunneling client.

Clientless Access

This topic describes the clientless access mode of WebVPN.



Clientless Mode Access

Clients with:

- Microsoft Windows 2000 or XP
- Linux

SNRS v2.0—4-5

In clientless mode, the remote user accesses the internal or corporate network using a web browser on the client machine. The PC of the remote user must run the Windows 2000, Windows XP, or Linux operating systems.

Benefits

Here are some of the benefits of clientless access:

- Clientless web-based access allows anywhere access to network resources.
- Web content transformation provides compatibility with web pages containing HTML and JavaScript.
- There is uniform and efficient application delivery via fully clientless Citrix support.
- These applications are supported: Intranet (HTML and JavaScript), Citrix, Windows file share (Common Internet File System [CIFS]).
- Multiple browser support ensures broad connection compatibility.

Applications supported in clientless mode include the following:

- Web browsing
  - — Using HTTP and HTTPS
  - — Provides a URL box and a list of web server links in the portal page that allows the remote user to browse the internal web sites

- File sharing
  - Using CIFS
  - Provides a list of file server links in the portal page that allows the remote user to do the following operations:
    - Browse a network (listing of domains)
    - Browse a domain (listing of servers)
    - Browse a server (listing of shares)
    - List the files in a share
    - Create a new file
    - Create a directory
    - Rename a directory
    - Update a file
    - Download a file
    - Remove a file
    - Rename a file

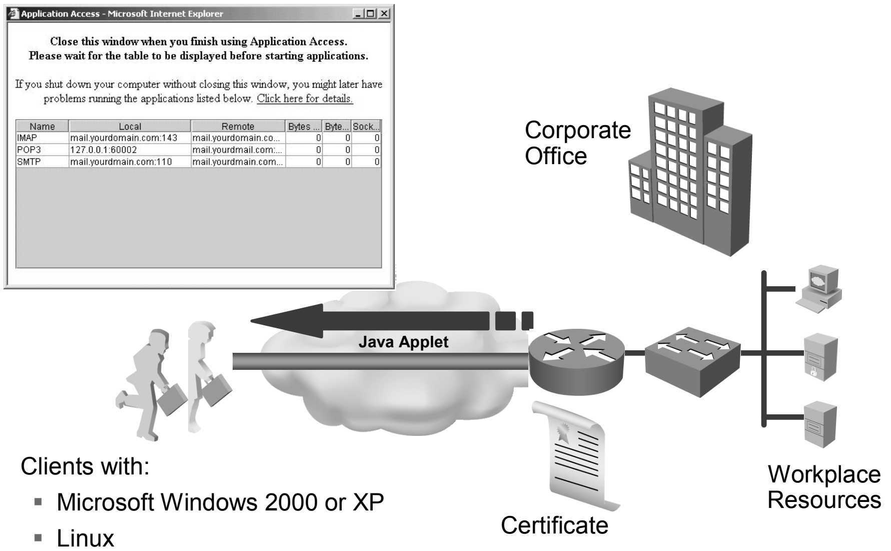
Note Linux requires that the Samba application be installed before CIFS file shares can be remotely accessed.

- Web-based e-mail
  - Such as Microsoft Outlook Web Access (OWA) 2003 (using HTTP and HTTPS) with Web-Based Distributed Authoring and Versioning (WebDAV) extensions
  - Provides a link that allows the remote user to connect to the exchange server and read web-based e-mail

Thin-Client Access

This topic describes the thin-client or port-forwarding access mode of WebVPN.

Thin-Client Mode Access



Application Access - Microsoft Internet Explorer

Close this window when you finish using Application Access.
Please wait for the table to be displayed before starting applications.

If you shut down your computer without closing this window, you might later have problems running the applications listed below. Click here for details.

Name	Local	Remote	Bytes ...	Byte...	Sock...
IMAP	mail.yourdomain.com:143	mail.yourdomain.co...	0	0	0
POP3	127.0.0.1:60002	mail.yourdmail.com:...	0	0	0
SMTP	mail.yourdmain.com:110	mail.yourdmain.com...	0	0	0

Clients with:

- Microsoft Windows 2000 or XP
- Linux

SNRS v2.0—4-6

Thin-client mode, also called TCP port forwarding, assumes that the client application uses TCP to connect to a well-known server and port.

In thin-client mode, the remote user downloads a Java applet by clicking the link provided on the portal page. The Java applet acts as a TCP proxy on the client machine for the services that you configure on the gateway.

The applications that are supported in thin-client mode are mainly e-mail-based (SMTP, POP3, and IMAP version 4 [IMAP4]) applications or other client-initiated TCP nonweb-based applications that use static ports.

The Java applet initiates an HTTP request from the remote user client to the WebVPN gateway. The name and port number of the internal e-mail server is included in the HTTP request (power-on self-test [POST] or CONNECT). The WebVPN gateway creates a TCP connection to that internal e-mail server and port.

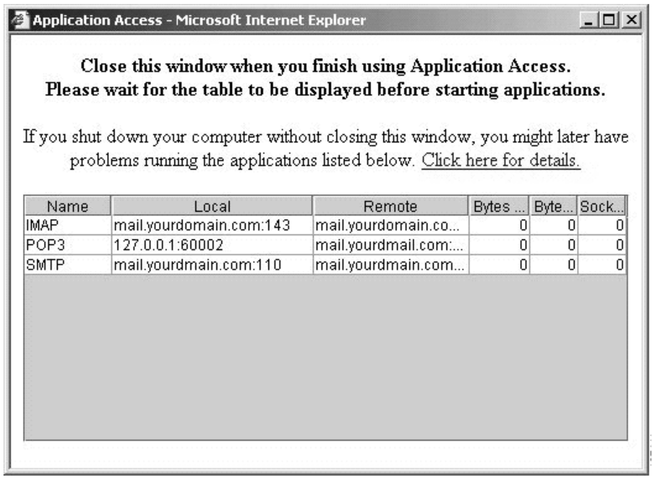
The Java applet starts a new SSL connection for every client connection.

Note	The TCP port-forwarding proxy works only with the Sun MicroSystems JRE version 1.4 or later releases. A Java applet is loaded through the browser that verifies the JRE version. The Java applet will refuse to run if a compatible JRE version is not detected.

Example

The figure shows an example of the port-forwarding application screen.

Application Access (Port-Forwarding) Screen



Close this window when you finish using Application Access.
Please wait for the table to be displayed before starting applications.

If you shut down your computer without closing this window, you might later have problems running the applications listed below. Click here for details.

Name	Local	Remote	Bytes ...	Byte...	Sock...
IMAP	mail.yourdomain.com:143	mail.yourdomain.co...	0	0	0
POP3	127.0.0.1:60002	mail.yourdmail.com:...	0	0	0
SMTP	mail.yourdmain.com:110	mail.yourdmain.com...	0	0	0

SNRS v2.0—4-7

The Java-based application helper supplements clientless access by providing connectivity to "non-webified" applications such as the following:

- POP, SMTP, or IMAP e-mail
- Instant messaging
- Calendar
- Client-initiated TCP-based applications such as Telnet

Restrictions

You should observe the following restrictions when using thin-client mode:

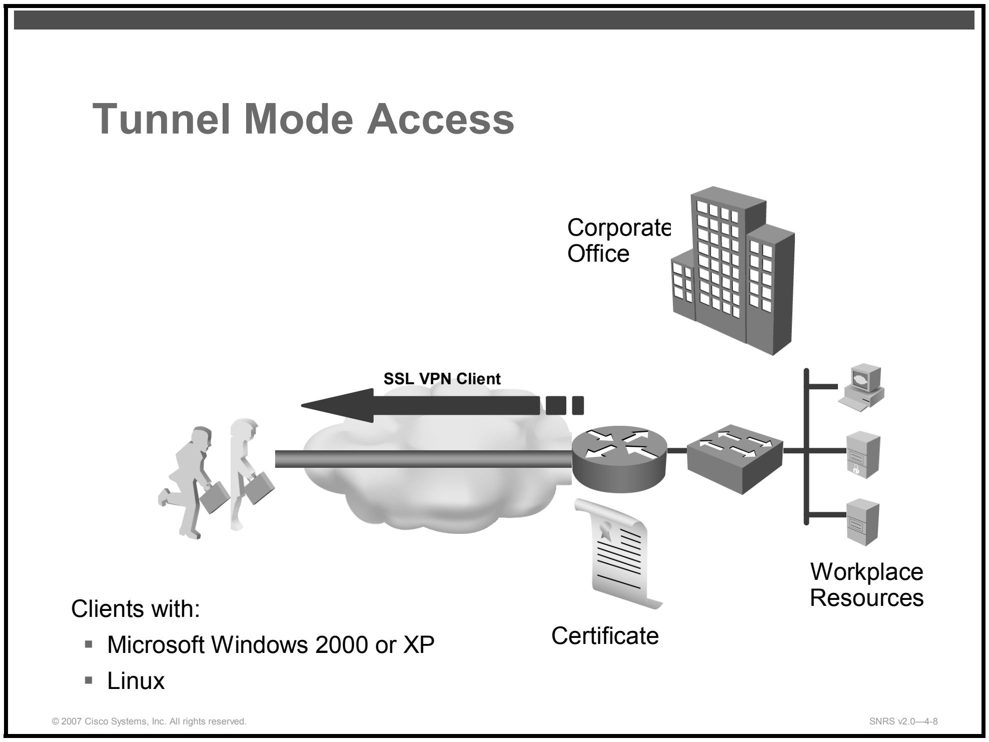
- The remote user must allow the Java applet to download and install.
- You cannot use thin-client mode for applications such as FTP where the ports are negotiated dynamically. You can use TCP port forwarding only with static ports.
- For applications to work seamlessly, you should give administrative privileges to remote users. If you do not give administrative privileges to remote users, remote users may need to manually change the client program settings so that the applications work properly.

Auto Applet Download

Effective with Cisco IOS Release 12.4(9)T, administrators have the option of automatically downloading the port-forwarding Java applet. This feature must be configured on a group policy basis.

Tunnel Mode Access

This topic describes the full network access mode of WebVPN.



Tunnel Mode Access

Clients with:

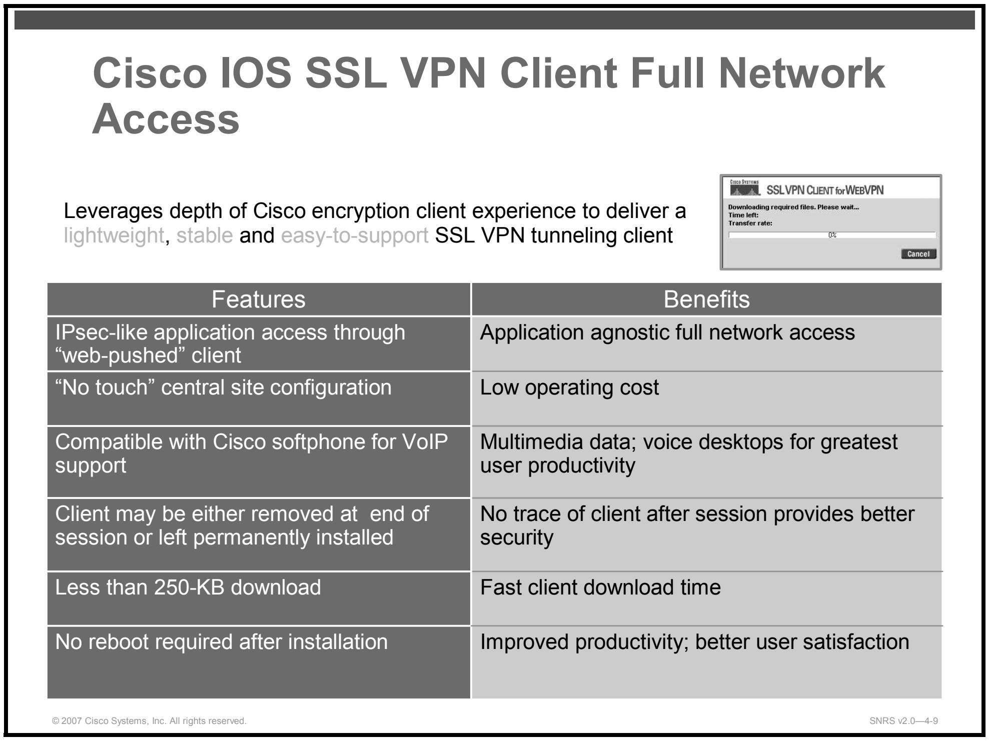
- Microsoft Windows 2000 or XP
- Linux

SNRS v2.0—4-8

In a typical clientless remote-access scenario, remote users establish an SSL tunnel to move data to and from the internal networks at the application layer (for example, web and e-mail). In tunnel mode, remote users use an SSL tunnel to move data at the network (IP) layer—IP over SSL. Therefore, tunnel mode supports most IP-based applications. Tunnel mode supports many popular corporate applications (for example, Microsoft Outlook, Microsoft Exchange, IBM Lotus Notes e-mail, FTP, Telnet, and so on).

The tunnel connection is determined by the group policy configuration. The Cisco SVC is downloaded (less than 250 KB) and installed to the remote user PC, and the tunnel connection is established when the remote user logs into the WebVPN gateway.

By default, the Cisco SVC is removed from the client PC after the connection is closed. However, you have the option to keep the Cisco SVC installed on the client PC.



Cisco IOS SSL VPN Client Full Network Access

Leverages depth of Cisco encryption client experience to deliver a lightweight, stable and easy-to-support SSL VPN tunneling client

Cisco Systems
SSL VPN Client for WebVPN
0%
Cancel

Downloading required files. Please wait...
Time left:
Transfer rate:

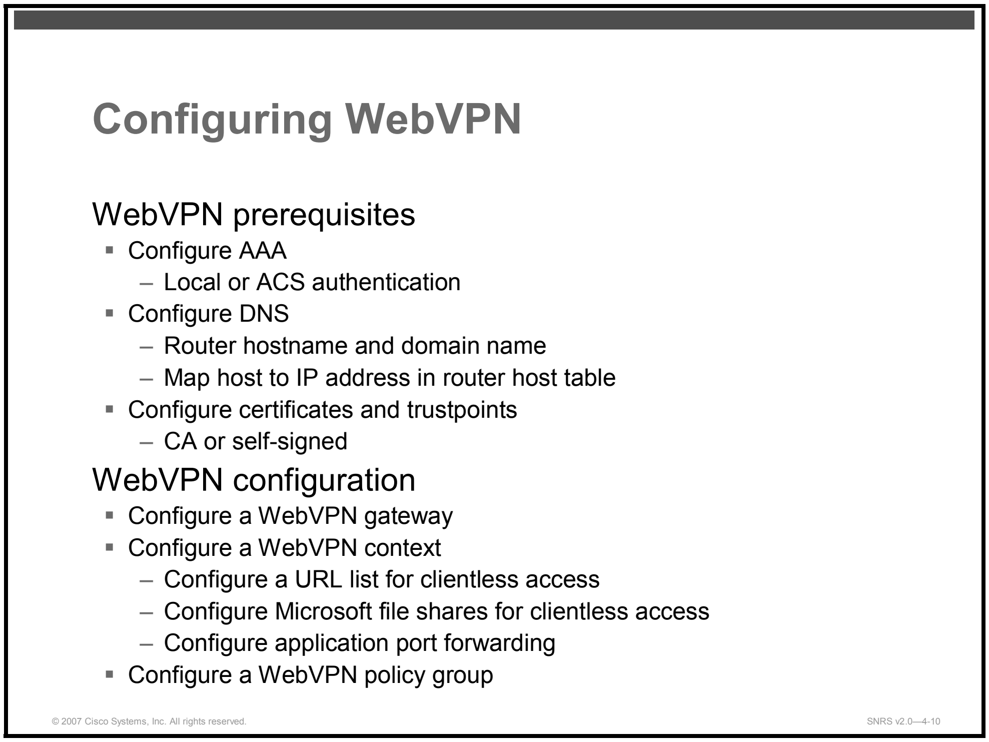
Features	Benefits
IPsec-like application access through “web-pushed” client	Application agnostic full network access
“No touch” central site configuration	Low operating cost
Compatible with Cisco softphone for VoIP support	Multimedia data; voice desktops for greatest user productivity
Client may be either removed at end of session or left permanently installed	No trace of client after session provides better security
Less than 250-KB download	Fast client download time
No reboot required after installation	Improved productivity; better user satisfaction

The Cisco SVC provides such features as the following:

- An IPsec-like application access through a “web-pushed” client
- “No touch” central site configuration
- VoIP support
- Client may be either removed at end of session or left permanently installed
- Less than 250-KB download
- No reboot required after installation

WebVPN Configuration Tasks

This topic lists the tasks required to configure WebVPN.



Configuring WebVPN

WebVPN prerequisites

- Configure AAA
  - Local or ACS authentication
- Configure DNS
  - Router hostname and domain name
  - Map host to IP address in router host table
- Configure certificates and trustpoints
  - CA or self-signed

WebVPN configuration

- Configure a WebVPN gateway
- Configure a WebVPN context
  - Configure a URL list for clientless access
  - Configure Microsoft file shares for clientless access
  - Configure application port forwarding
- Configure a WebVPN policy group

SNRS v2.0—4-10

This figure lists the basic tasks required to configure a Cisco IOS SSL VPN. There are several issues that must be addressed when setting up a Cisco IOS SSL VPN.

Before configuring WebVPN, an administrator must configure and install the following:

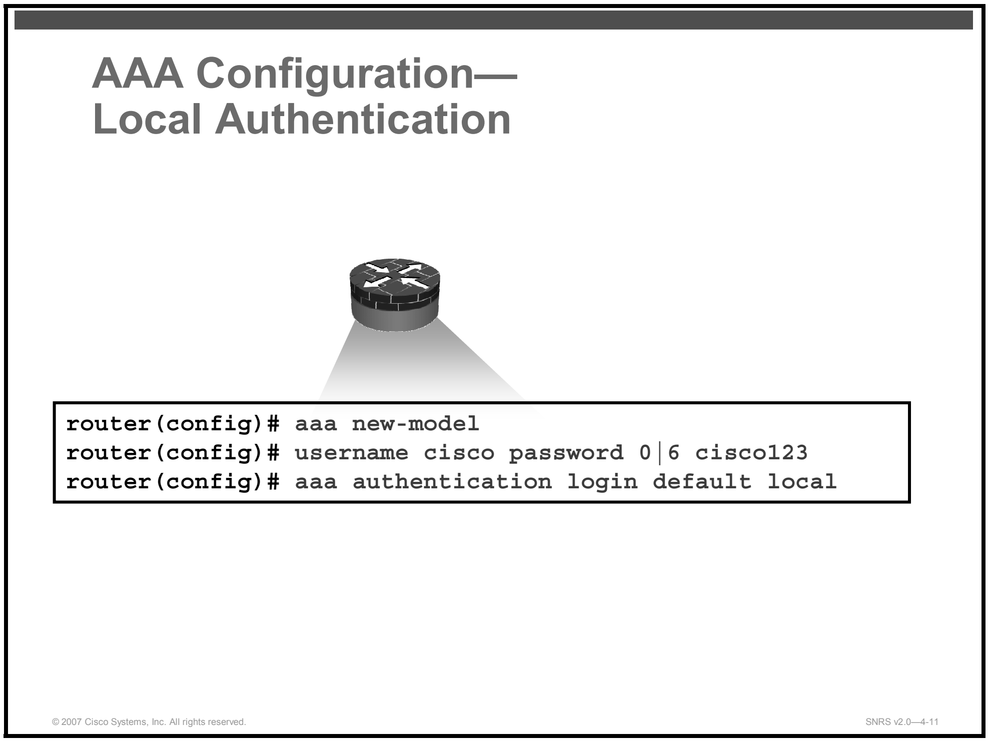
- **Authentication, authorization, and accounting (AAA):** This includes setting up local or remote authentication.
- **Domain Name System (DNS):** This includes configuring the hostname, domain name, and DNS name servers or defining a static hostname-to-address mapping in the host (router) cache.
- **Certificates and trustpoints:** This includes requesting and installing certificates and configuring trustpoints.

After meeting the prerequisites, you can then configure the WebVPN services:

- **WebVPN gateway:** This includes configuring the address, hostname, and trustpoint for the WebVPN.
- **WebVPN context:** This includes the ability to customize the user interface of the site and to configure such options as login message, URLs that will appear on the portal, and defining policy groups.

AAA Configuration for WebVPN

This topic describes how to configure AAA for WebVPN.



AAA Configuration—Local Authentication

```
router(config)# aaa new-model
router(config)# username cisco password 0|6 cisco123
router(config)# aaa authentication login default local
```

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-11

Before configuring WebVPN for a AAA-related configuration, an administrator must create user accounts using either local authentication or authentication via AAA (RADIUS and TACACS+ servers) and configure AAA-related commands.

Local Authentication

The figure lists the commands used to configure AAA local authentication. Follow this procedure to configure AAA local authentication:

Step 1 Enable the AAA access control model.

```
router(config)# aaa new-model
```

Step 2 Populate the local user database.

```
router(config)# username user1 password 0|6 cisco
```

SYNTAX:

```
router(config)# )# username <name> password 0|6 password <password>
```

Where:

- *name:* Name of user.
- **0:** Unencrypted password follows.
- **6:** Encrypted password follows.
- *Password:* User password.

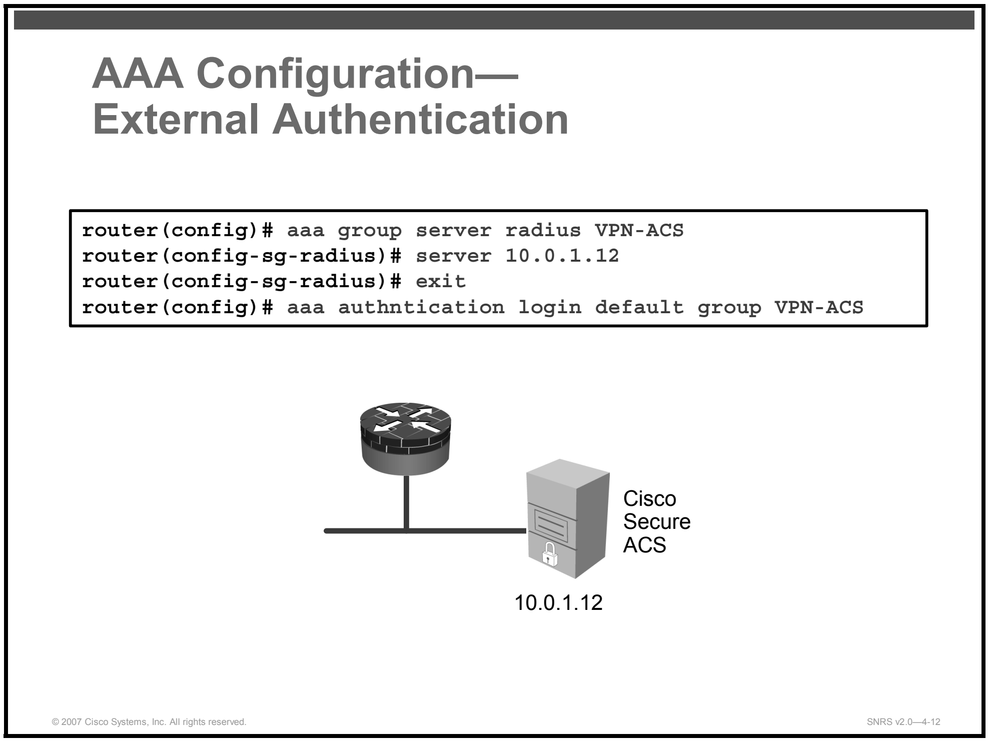
Step 3 Specify local AAA authentication.

```
router(config)# aaa authentication login default local
```

The database that is configured for remote-user authentication on the WebVPN gateway can be a local database, as shown here, or the database can be accessed through any RADIUS or TACACS+ AAA server.

External Authentication

Cisco recommends that you use a separate AAA server, such as a Cisco Secure Access Control Server (ACS). A separate AAA server provides a more robust security solution. It allows you to configure unique passwords for each remote user, and accounting and logging for remote-user sessions. This figure lists the commands that are used to configure AAA for remote authentication.



AAA Configuration—
External Authentication

```
router(config)# aaa group server radius VPN-ACS
router(config-sg-radius)# server 10.0.1.12
router(config-sg-radius)# exit
router(config)# aaa authntication login default group VPN-ACS
```

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-12

This section shows how to configure AAA using an external RADIUS server. AAA is configured in global configuration mode. The authentication method list is referenced in the WebVPN context configuration with the **aaa authentication** command. The steps in this task configure AAA using a RADIUS server.

Follow this procedure to configure AAA remote authentication:

Step 1 Enable the AAA access control model.

```
router(config)# aaa new-model
```

Step 2 Configure a RADIUS or TACACS+ server group and specify the authentication list or method.

```
router(config)# aaa group server radius VPN-ACS
```

SYNTAX:

```
router(config)# aaa group server {radius group-name | tacacs+ group-name}
```

Where:

- *group-name:* Specifies the name of the server group.

Step 3 Specify the IP address of the AAA group server.

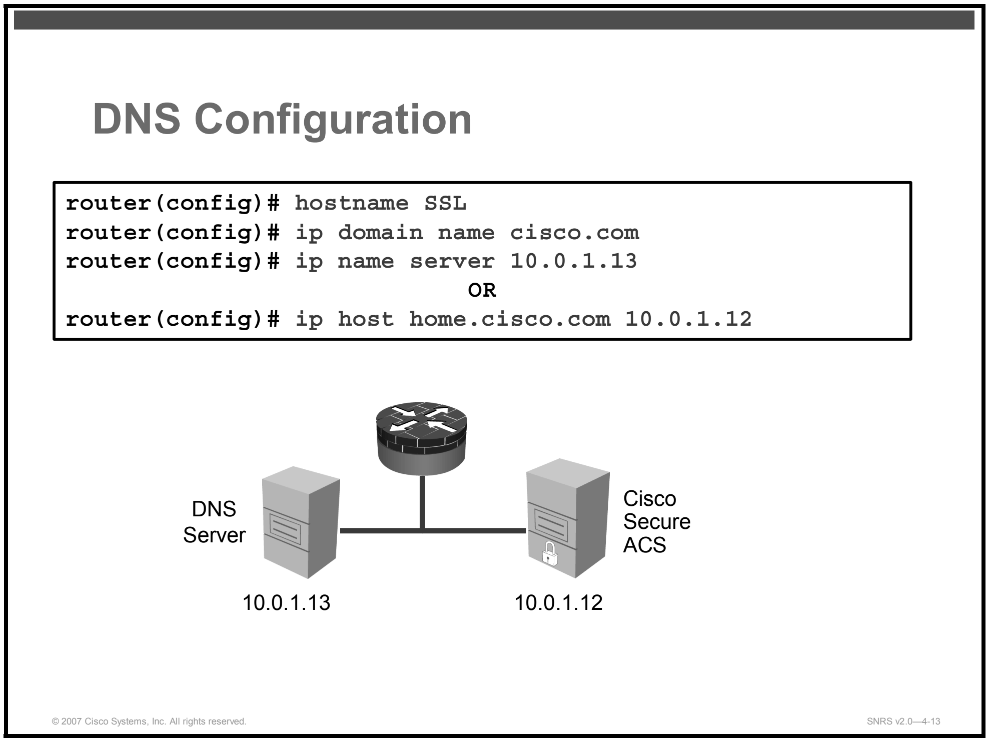
```
router(config-sg-radius)# server 10.1.1.20
```

Step 4 Set the AAA login parameters.

```
router(config)# aaa authentication login default group VPN-ACS
```

DNS Configuration for WebVPN

This topic describes how to configure DNS for WebVPN.



DNS Configuration

```
router(config)# hostname SSL
router(config)# ip domain name cisco.com
router(config)# ip name server 10.0.1.13
                                OR
router(config)# ip host home.cisco.com 10.0.1.12
```

SNRS v2.0—4-13

Before configuring WebVPN, an administrator must configure DNS-related commands. The hostname and the domain name must be set as well as any name servers that may be in use.

The following commands are used to configure DNS parameters for use with WebVPN:

Step 1 Specify a hostname for the router.

```
router(config)# hostname SSL
```

```
SYNTAX:
router(config)# hostname name
Where:
```

- *name:* New hostname for the network device

Step 2 Define a default domain name.

```
router(config)# ip domain name cisco.com
```

```
SYNTAX:
router(config)# ip domain name name
Where:
```

- *name* : Default domain name used to complete unqualified hostnames

Note Do not include the initial period that separates an unqualified name from the domain name.

You can point to your DNS server or populate the local router table with hostname-to-IP address mappings.

When using a name server, follow this additional step:

Step 3 Specify the address of one or more name servers.

```
router(config)# ip name server 10.0.1.13
```

```
SYNTAX:
```

```
router(config)# ip name-server [vrf vrf-name] server-address1 [server-
address2...server-address6]
```

```
Where:
```

- *server-address1:* IPv4 or IPv6 addresses of a name server.
- *server-address2...server-address6:* (Optional) IP addresses of additional name servers (a maximum of six name servers).

When using static mappings, follow this additional step:

Step 4 Define a static hostname-to-address mapping on the router.

```
router(config)# ip host home.cisco.com 10.0.1.12
```

```
SYNTAX:
```

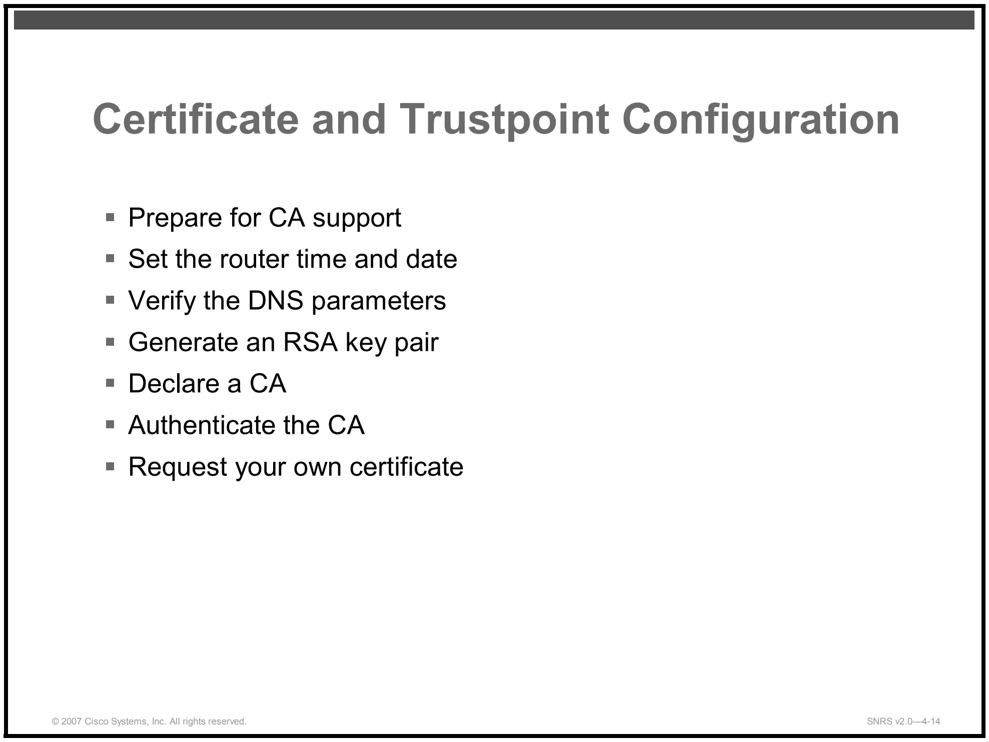
```
router(config)# ip host [vrf vrf-name] {name | tmodem-telephone-
number} [tcp-port-number] address1 [address2...address8]
```

```
Where:
```

- **vrf** *vrf-name:* (Optional) This defines a VPN)routing and forwarding VRF table. The *vrf-name* argument specifies a name for the VRF table.
- *name:* This is the name of the host. The first character can be either a letter or a number. If you use a number, the types of operations you can perform are limited.
- **t***modem-telephone-number:* This is the modem telephone number that is mapped to the IP host address for use in Cisco modem user interface mode. You must enter the letter **t** before the telephone number.
- *tcp-port-number:* (Optional) This is the TCP port number to connect to when using the defined hostname in conjunction with an EXEC connect or Telnet command. The default is Telnet (port 23).
- *address1:* This is the associated IP host address.
- *address2...address8:* (Optional) This is for additional associated IP addresses. You can bind up to eight addresses to a hostname.

Certificates and Trustpoints for WebVPN

This topic describes the process for configuring certificates and trustpoints for WebVPN.



Certificate and Trustpoint Configuration

- Prepare for CA support
- Set the router time and date
- Verify the DNS parameters
- Generate an RSA key pair
- Declare a CA
- Authenticate the CA
- Request your own certificate

WebVPN is based on HTTPS, which requires a public key infrastructure (PKI) trustpoint to be configured. A self-signed certificate is automatically generated when a WebVPN gateway is put in service. However, if network security policy dictates that you use an external certificate authority (CA) server, use the discussion here as a guide.

The figure lists the tasks involved with configuring certificates and trustpoints for use with a Cisco IOS SSL VPN. The process is the same as for a site-to-site IPsec VPN discussed in "Implementing IPsec VPNs Using PKI" lesson. Network security policy should determine which method is used.

To configure certificates and trustpoints to use with Cisco WebVPN, follow these steps:

Step 1 Prepare for CA support by gathering the information of the CA server such as IP address, server name, and URL.

Step 2 Make sure that the router date and time are set correctly for your time zone.

Note One of the first things checked within a certificate is the expiration. A valid date is required, and the router has to have the correct time.

Step 3 Verify the router DNS parameters and add the CA server to the router host table by using the **ip host** command.

Step 4 Generate your RSA key pair as described previously using the **crypto key generate rsa** command.

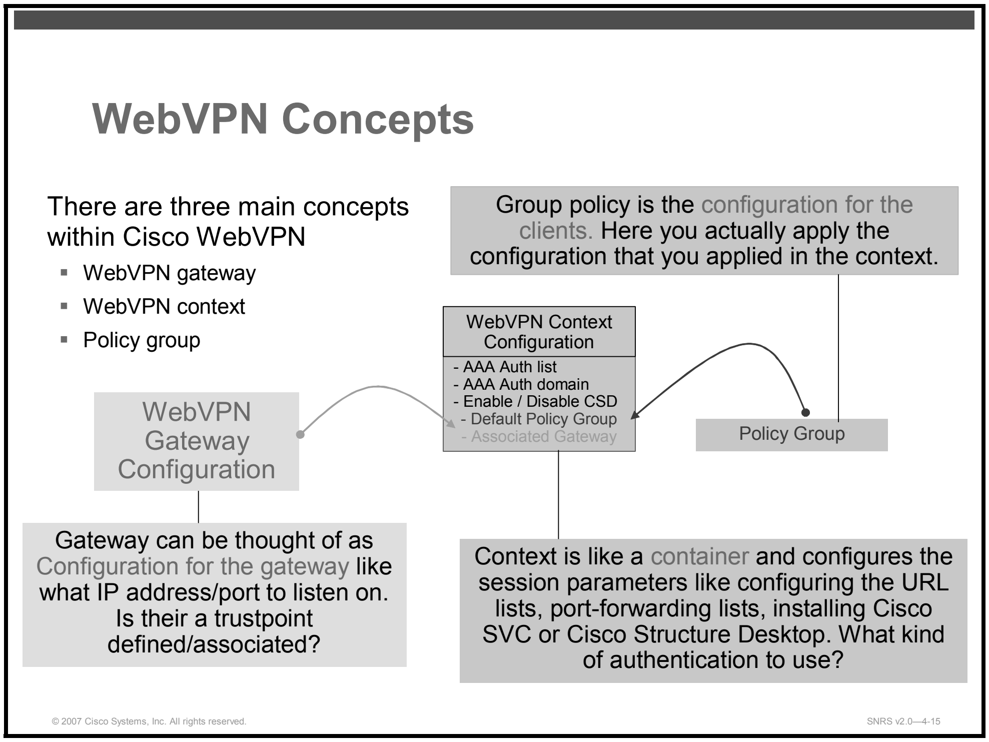
Step 5 Declare the CA using the **crypto pki trustpoint** command.

Step 6 Authenticate the CA using the **crypto pki authenticate** command.

Step 7 Request your own certificate using the **crypto pki enroll** command.

WebVPN Configuration

The topic describes how to configure WebVPN services on a router.



WebVPN Concepts

There are three main concepts within Cisco WebVPN

- WebVPN gateway
- WebVPN context
- Policy group

Group policy is the configuration for the clients. Here you actually apply the configuration that you applied in the context.

Gateway can be thought of as Configuration for the gateway like what IP address/port to listen on. Is their a trustpoint defined/associated?

Context is like a container and configures the session parameters like configuring the URL lists, port-forwarding lists, installing Cisco SVC or Cisco Structure Desktop. What kind of authentication to use?

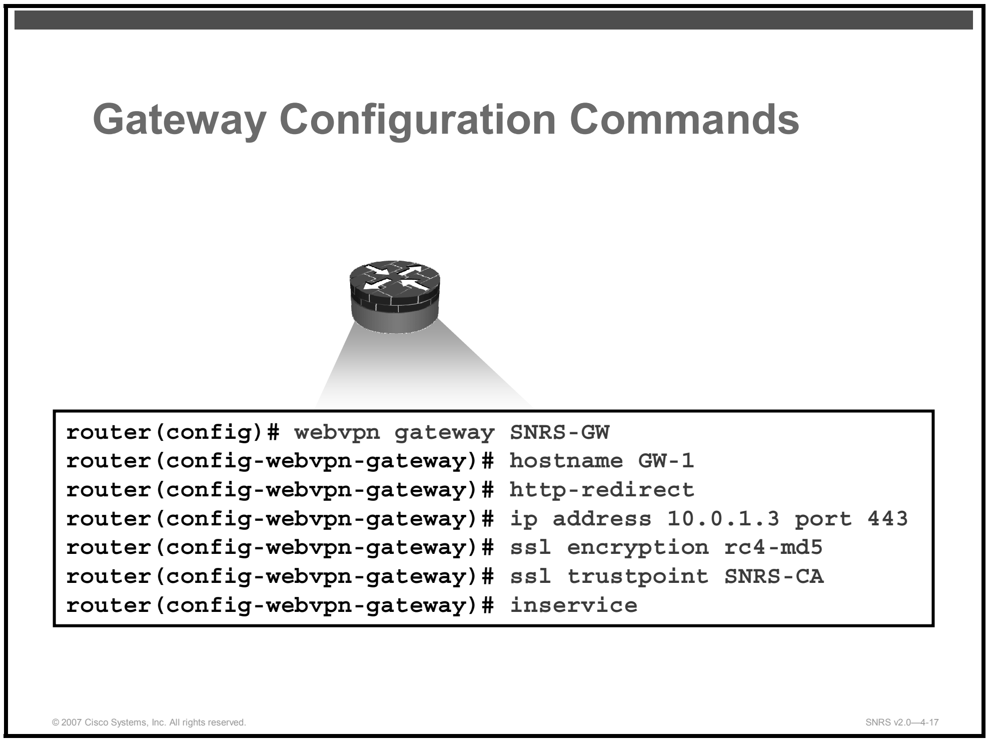
SNRS v2.0—4-15

There are three main components to configure within WebVPN.

Configuring a basic WebVPN portal includes the following tasks:

- **Configure the WebVPN virtual gateway:** Before using the WebVPN feature, a virtual gateway must be configured and put in service. This specifies the IP address and port to use for WebVPN and configures the trustpoint to use. The IP address should be a public IP address configured on an interface or loopback interface on the WebVPN gateway. The default port is 443, and the default trustpoint name is “SSLVPN.” By putting the virtual gateway in service, WebVPN service is enabled on the gateway.
- **Configure WebVPN virtual context:** A WebVPN virtual context must be configured to associate the virtual WebVPN gateway with the configured features. Multiple virtual contexts can be configured on the secure gateway, giving access to various features and access modes, depending on the domain configured for each context.
- **Configure URL lists:** A URL box is provided on the portal page to allow web browsing. Lists of web server links can be configured and will be displayed on the portal page. These URL lists are provided for ease of navigating the internal websites. The URL lists are configured in the WebVPN context submode, and must be defined in the group policy for the given WebVPN context. If you have enabled Citrix in the group policy, you can add a link in the URL list to the Citrix server.

- **Configure WebVPN group policies:** A group policy is configured for each WebVPN virtual instance. The group policy specifies the WebVPN features and parameters to be used for this virtual instance. The Citrix, CIFS, Cisco Secure Desktop, thin-client mode, and full tunnel mode features can be enabled or disabled in the group policy, which is then associated with the WebVPN context.



Gateway Configuration Commands

```
router(config)# webvpn gateway SNRS-GW
router(config-webvpn-gateway)# hostname GW-1
router(config-webvpn-gateway)# http-redirect
router(config-webvpn-gateway)# ip address 10.0.1.3 port 443
router(config-webvpn-gateway)# ssl encryption rc4-md5
router(config-webvpn-gateway)# ssl trustpoint SNRS-CA
router(config-webvpn-gateway)# inservice
```

SNRS v2.0—4-17

Configuring the Virtual Gateway

The WebVPN gateway will act as a proxy for connections to protected corporate resources. Protected resources are accessed through an SSL-encrypted connection between the gateway and a web-enabled browser on a remote client. You will use the **webvpn gateway** command to place the router in Cisco IOS SSL VPN gateway configuration mode.

- Only one gateway is configured in a Cisco IOS WebVPN-enabled network.
- The configuration of the **ssl trustpoint** command is required only if you need to configure a specific CA certificate. A self-signed certificate is automatically generated when a WebVPN gateway is put in service.

Follow these steps to configure a WebVPN gateway:

Step 1 Name the gateway and enter SSL VPN gateway configuration mode.

```
router(config)# webvpn gateway SNRS-GW
```

SYNTAX:

```
router(config)# webvpn gateway name
```

Step 2 Specify the hostname for the WebVPN gateway.

```
router(config-webvpn-gateway)# hostname GW-1
```

SYNTAX:

```
router(config-webvpn-gateway)# hostname name
```

Step 3 Configure HTTP traffic to be carried over HTTPS.

```
router(config-webvpn-gateway)# http-redirect
```

SYNTAX:

```
router(config-webvpn-gateway)# http-redirect [port number]
```

Note When this command is enabled, the WebVPN gateway listens on port 80 and redirects HTTP traffic over port 443 or the port number specified with the **port** keyword.

Step 4 Configure a proxy IP address for the WebVPN gateway.

```
router(config-webvpn-gateway)# ip address 10.0.1.2 port 443
```

SYNTAX:

```
router(config-webvpn-gateway)# ip address number [port number]
[secondary]
```

- A secondary address must be configured if the proxy IP address is not on a directly connected network.
- A secondary address does not reply to Address Resolution Protocol (ARP) or Internet Control Message Protocol (ICMP) messages.

Step 5 Specify the encryption algorithm that the SSL protocol will use for Cisco IOS SSL VPN connections.

```
router(config-webvpn-gateway)# ssl encryption [3des-sha1]
[aes-sha1] [rc4-md5]
```

Note The Cisco IOS SSL VPN provides remote-access connectivity from almost any Internet-enabled location using only a web browser and its native SSL encryption. The **ssl encryption** command is configured to restrict the encryption algorithms that SSL uses in Cisco IOS Software.

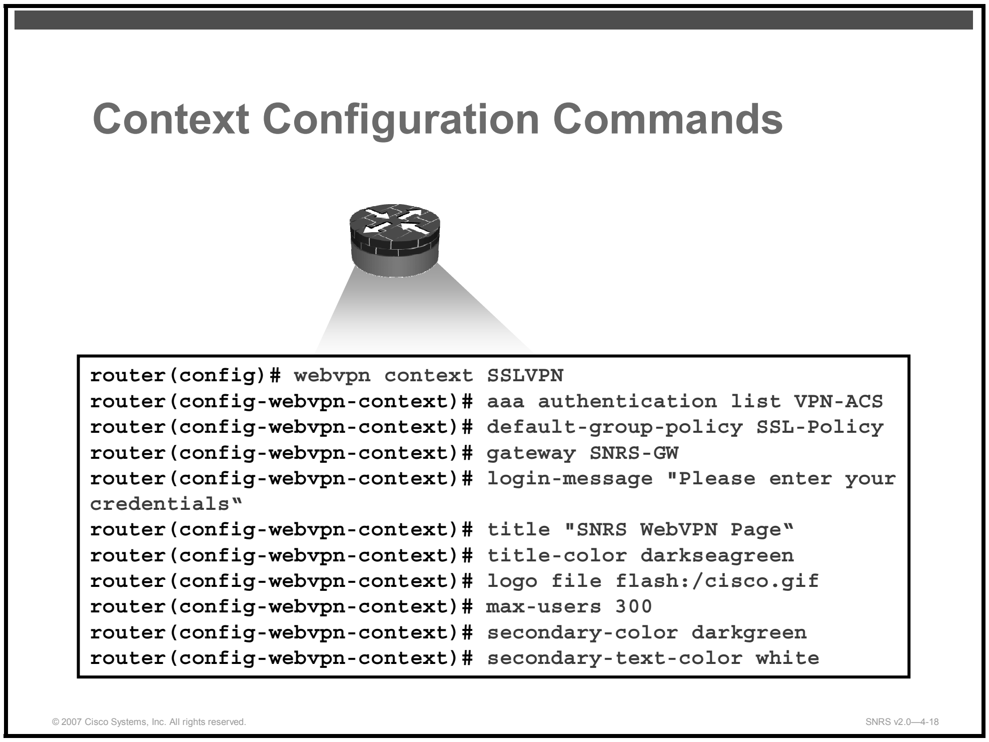
Step 6 Configure the certificate trustpoint for the WebVPN gateway.

```
router(config-webvpn-gateway)# ssl trustpoint SNRS-CA
```

Note Entering the **no** form of this command configures the WebVPN gateway to revert to using an autogenerated self-signed certificate.

Step 7 Enable the WebVPN gateway.

```
router(config-webvpn-gateway)# inservice
```



Context Configuration Commands

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# aaa authentication list VPN-ACS
router(config-webvpn-context)# default-group-policy SSL-Policy
router(config-webvpn-context)# gateway SNRS-GW
router(config-webvpn-context)# login-message "Please enter your
credentials‟
router(config-webvpn-context)# title "SNRS WebVPN Page‟
router(config-webvpn-context)# title-color darkseagreen
router(config-webvpn-context)# logo file flash:/cisco.gif
router(config-webvpn-context)# max-users 300
router(config-webvpn-context)# secondary-color darkgreen
router(config-webvpn-context)# secondary-text-color white
```

Configuring the Virtual Context

The WebVPN context defines the virtual configuration of the Cisco IOS SSL VPN. Entering the **webvpn context** command places the router in Cisco IOS SSL VPN configuration mode.

Follow these steps to configure a WebVPN context:

Step 1 Enter Cisco IOS SSL VPN configuration mode to configure the WebVPN context.

```
router(config)# webvpn context name
```

If you have not already set up your WebVPN context, this will create a new context and enter Cisco IOS SSL VPN configuration mode.

Step 2 Configure AAA authentication for Cisco IOS SSL VPN sessions.

```
router(config-webvpn-context)# aaa authentication {domain name
| list name}
```

Syntax Description

domain *name*	Configures authentication using the specified domain name
list *name*	Configures authentication using the specified list name

Note If this command is not configured, the WebVPN gateway will use global AAA parameters (if configured) for remote-user authentication.

Step 3 Associate a group policy with the WebVPN context configuration.

```
router(config-webvpn-context)# default-group-policy name
```

This command is configured to attach the policy group to the WebVPN context when multiple group policies are defined under the context.

This policy will be used as the default, unless a AAA server pushes an attribute that specifically requests another group policy.

Step 4 Associate a WebVPN gateway with this WebVPN context.

```
router(config-webvpn-context)# gateway name [domain name |
virtual-host name]
```

The gateway configured in Step 1 is associated with the WebVPN context in this configuration step.

Step 5 Enable the WebVPN context configuration.

```
router(config-webvpn-context)# inservice
```

The context is put in service by entering this command. However, the context is not operational until it is associated with an enabled WebVPN gateway.

Step 6 Configure a message for the user login text box displayed on the login page.

```
router(config-webvpn-context)# login-message [message-string]
```

Step 7 Configure the HTML title string that is shown in the browser title and on the title bar of a Cisco IOS SSL VPN.

```
router(config-webvpn-context)# title [title-string]
```

The title of the page should reflect the business needs of the company.

Step 8 (Optional) Specify the color of the title bars on the login and portal pages of a Cisco IOS SSL VPN. This example shows the three forms that can be used to configure the title color.

```
router(config-webvpn-context)# title-color color
```

The value for the color argument is entered as a comma-separated (red, green, blue [RGB]) value (CSV), an HTML color value (beginning with a pound sign [#]), or the name of the color that is recognized in HTML (no spaces between words or characters). The CSV is limited to 32 characters. The CSV is parsed to ensure that it matches one of the following formats (using Perl regex notation). The default color is purple.

Step 9 (Optional) Configure a custom logo to be displayed on the login and portal pages of a Cisco IOS SSL VPN.

```
router(config-webvpn-context)# logo [file filename | none]
```

The source image file for the logo is a .gif, .jpg, or .png file that is up to 255 characters in length (filename) and up to 100 KB in size. The file is referenced from a local file system, such as flash memory. An error message will be displayed if the file is not referenced from a local file system. No logo will be displayed if the image file is removed from the local file system.

Step 10 (Optional) Limit the number of connections to a Cisco IOS SSL VPN that will be permitted.

```
router(config-webvpn-context)# max-users number
```

Step 11 (Optional) Configure the color of the secondary title bars on the login and portal pages of a Cisco IOS SSL VPN.

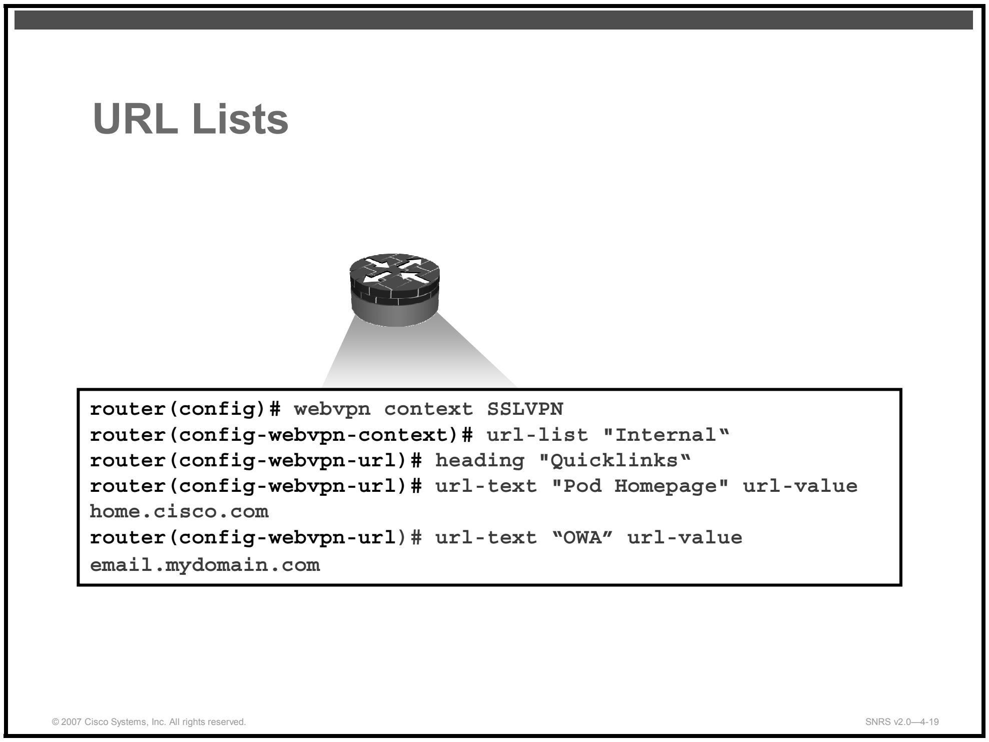
```
router(config-webvpn-context)# secondary-color color
```

The value for the color argument is entered as a CSV RGB, an HTML color value (beginning with a pound sign [#]), or the name of the color that is recognized in HTML (no spaces between words or characters). The value is limited to 32 characters. The value is parsed to ensure that it matches one of the following formats (using Perl regex notation). The default color is purple.

Step 12 (Optional) Configure the color of the text on the secondary bars of a Cisco IOS SSL VPN.

```
router(config-webvpn-context)# secondary-text-color {black |
white}
```

The color of the text on the secondary bars must be aligned with the color of the text on the title bar. The default color is black.



URL Lists

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# url-list "Internal“
router(config-webvpn-url)# heading "Quicklinks“
router(config-webvpn-url)# url-text "Pod Homepage" url-value
home.cisco.com
router(config-webvpn-url)# url-text “OWA” url-value
email.mydomain.com
```

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-19

Creating URL Lists

The URL list is a list of HTTP URLs that are displayed on the portal page after a successful login. A URL box is provided on the portal page to allow web browsing. Lists of web server links can be configured and will be displayed on the portal page. These URL lists are provided for ease of navigating the internal websites. The URL lists are configured in the Cisco IOS WebVPN context submode, and must be defined in the group policy for the given Cisco IOS WebVPN context. If you have enabled Citrix in the group policy, you can add a link in the URL list to the Citrix server.

These steps show how to configure a URL list.

Step 1 Enter Cisco IOS SSL VPN configuration mode.

```
router(config)# webvpn context context-name
```

Step 2 Enter the Cisco IOS SSL VPN URL list configuration mode to configure the list of URLs to which a user has access on the portal page of a Cisco IOS SSL VPN.

```
router(config-webvpn-context)# url-list name
```

Step 3 Configure the heading that is displayed above URLs listed on the portal page of a Cisco IOS SSL VPN.

```
router(config-webvpn-url)# heading "Quicklinks"
```

SYNTAX:

```
router(config-webvpn-url)# heading text-string
```

The URL list heading is entered as a text string. The heading must be entered inside of quotation marks if it contains spaces.

Step 4 Add an entry to the URL list.

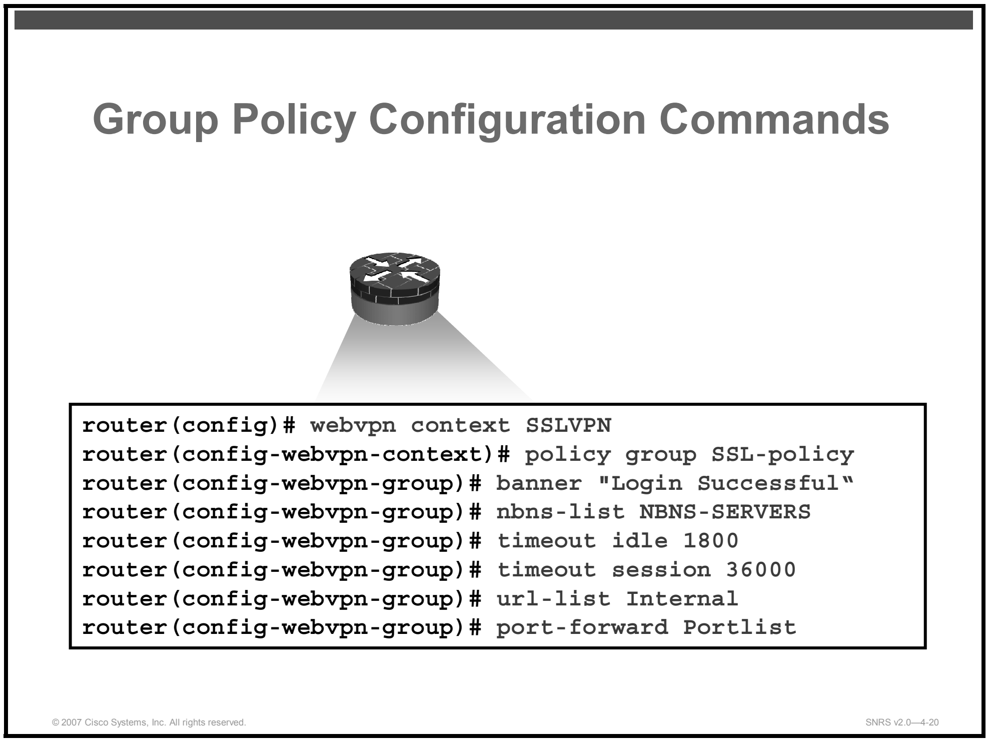
```
router(config-webvpn-url)# url-text { name url-value url}
```

URL Lists—Examples

It is easy to customize the portal page by separating functions or sites by URL lists. The following example creates two URL lists—one for the internal web and another for Citrix functions.

```
!
url-list “Internal web”
heading “Internal”
url-text “OWA” url-value “email.mydomain.com”
url-text “Homepage” url-value “www.mydomain.com”
!
url-list “Citrix”
heading “Citrix”
url-text “Citrix Server” url-value http://citrix-server.mydomain.com
```

Once the URL lists have been built, they must be associated with the WebVPN context by configuring it in the context policy group. The next section covers the configuration of a policy group.



Group Policy Configuration Commands

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# policy group SSL-policy
router(config-webvpn-group)# banner "Login Successful"
router(config-webvpn-group)# nbns-list NBNS-SERVERS
router(config-webvpn-group)# timeout idle 1800
router(config-webvpn-group)# timeout session 36000
router(config-webvpn-group)# url-list Internal
router(config-webvpn-group)# port-forward Portlist
```

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-20

Configuring the Group Policy

The policy group is a container that defines the presentation of the portal and the permissions for resources that are configured for a group of remote users. Entering the **policy group** command places the router in SSL VPN group policy configuration mode. After it is configured, the group policy is attached to the WebVPN context configuration by configuring the **default-group-policy** command.

Follow these steps to create a WebVPN group policy:

Step 5 Enter SSL VPN configuration mode.

```
router(config)# webvpn context SSLVPN
```

Step 6 Enter SSL VPN group policy configuration mode.

```
router(config-webvpn-context)# policy group name
```

Step 7 Configure a banner to be displayed after a successful login.

```
router(config-webvpn-group)# banner string
```

Where:

- *string*: Text string that contains 7-bit ASCII values and HTML tags and escape sequences; the text banner must be in quotation marks if it contains spaces.

Step 8 Attach a NetBIOS Name Service (NBNS) server list to this policy group configuration.

```
router(config-webvpn-group)# nbns-list name
```

Note The NBNS server list is first defined in SSL VPN NBNS list configuration mode.

Step 9 Configure remote user session idle time or the total length of time that a session can remain connected.

```
router(config-webvpn-group)# timeout {idle seconds | session
seconds}
```

Where:

- **idle** *seconds:* Configures the length of time that an end-user connection can remain idle
- **session** *seconds:* Configures the total length of time that an end user can maintain a single connection

Step 10 Attach a URL list to this policy group configuration.

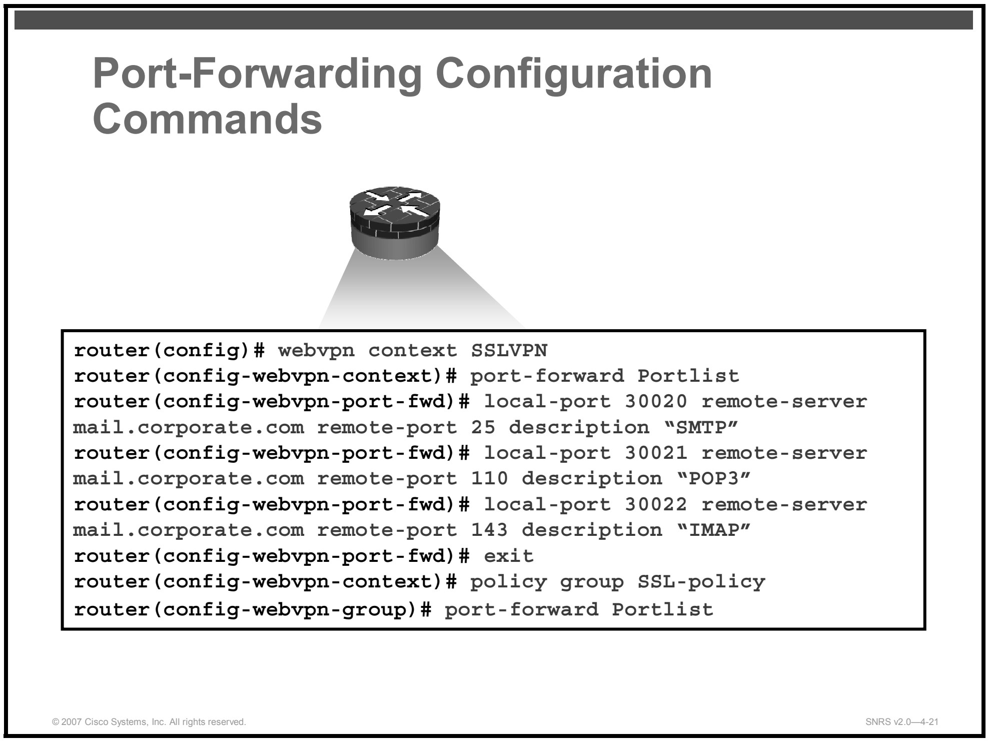
```
router(config-webvpn-group)# url-list name
```

Where:

- *name:* Name of the URL list configured in SSL VPN context configuration mode

Step 11 Attach a port-forwarding list to this group policy. This command is used when configuring the thin-client access mode.

```
router(config-webvpn-group)# port-forward portlist-name
```



Port-Forwarding Configuration Commands

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# port-forward Portlist
router(config-webvpn-port-fwd)# local-port 30020 remote-server
mail.corporate.com remote-port 25 description “SMTP”
router(config-webvpn-port-fwd)# local-port 30021 remote-server
mail.corporate.com remote-port 110 description “POP3”
router(config-webvpn-port-fwd)# local-port 30022 remote-server
mail.corporate.com remote-port 143 description “IMAP”
router(config-webvpn-port-fwd)# exit
router(config-webvpn-context)# policy group SSL-policy
router(config-webvpn-group)# port-forward Portlist
```

SNRS v2.0—4-21

Configuring Thin-Client Mode (TCP Port Forwarding)

The **port-forward** command is used to create the port-forwarding list. Application port number mapping (port forwarding) is configured with the **local-port** command in SSL VPN port-forward configuration mode.

A port-forwarding list is configured for thin-client mode WebVPN. Port forwarding extends the cryptographic functions of the SSL-protected browser to provide remote access to TCP and User Datagram Protocol (UDP)-based applications that use well-known port numbers, such as POP3, SMTP, IMAP, Telnet, SSH, and so on.

When port forwarding is enabled, the hosts file on the WebVPN client is modified to map the application to the port number configured in the forwarding list. The application port mapping is restored to the default when the user terminates the WebVPN session.

The following steps are required to configure the WebVPN for thin-client mode (also known as TCP port forwarding):

Step 1 Enter SSL VPN configuration mode.

```
router(config)# webvpn context SSLVPN
```

Step 2 Name a port-forwarding list and enter SSL VPN port-forwarding list configuration mode.

```
router(config-webvpn-context)# port-forward name
```

Syntax Description

name	Name of the port-forwarding list

Step 3 Remap (forward) application port numbers in the port-forwarding list.

```
router(config-webvpn-port-fwd)# local-port {number remote-
server name remote-port number description text-string}
```

Syntax Description

number	Configures the port number to which the local application is mapped A number from 1 through 65535 is entered.
remote-server *name*	Identifies the remote server An IP version 4 (IPv4) address or fully qualified domain name is entered.
remote-port *number*	Specifies the well-known port number of the application, for which port forwarding is to be configured; A number from 1 through 65535 is entered.
description *text-string*	Configures a description for this entry in the port-forwarding list The text string is displayed on the end-user applet window. A text string of up to 64 characters in length is entered.

Step 4 Exit SSL VPN port-forwarding list configuration mode, and enter SSL VPN configuration mode.

```
router(config-webvpn-port-fwd)# exit
```

Step 5 Enter SSL VPN group policy configuration mode.

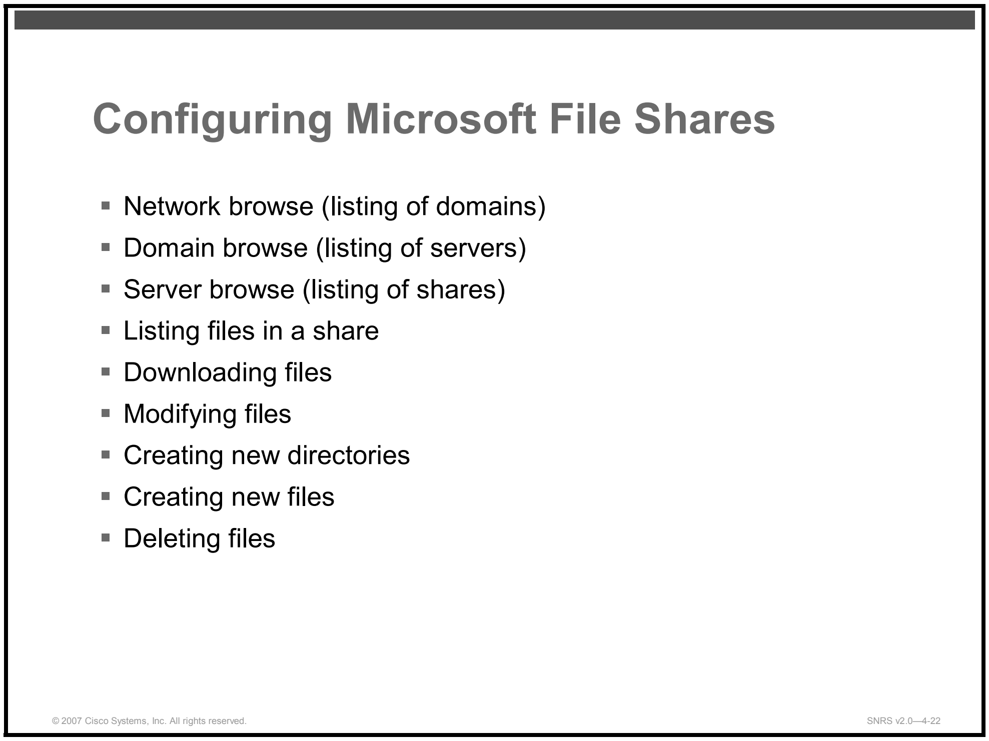
```
router(config-webvpn-context)# policy group SSL-policy
```

Step 6 Attach a port-forwarding list to this policy group configuration.

```
router(config-webvpn-group)# port-forward name [auto-download]
```

Where:

- *name:* Name of the port-forwarding list configured in SSL VPN configuration mode
- **auto-download:** (Optional) Allows for automatic download of the port-forwarding Java applet on the portal page of a website



Configuring Microsoft File Shares

- Network browse (listing of domains)
- Domain browse (listing of servers)
- Server browse (listing of shares)
- Listing files in a share
- Downloading files
- Modifying files
- Creating new directories
- Creating new files
- Deleting files

SNRS v2.0—4-22

Configuring Microsoft File Shares for Clientless Remote Access

In clientless remote-access mode, files and directories created on Microsoft Windows servers can be accessed by the remote client through the HTTPS-enabled browser. When enabled, a list of file server and directory links are displayed on the portal page after login. The administrator can customize permissions on the WebVPN gateway to provide limited read-only access for a single file or full write access and network browsing capabilities.

CIFS Support

CIFS is the protocol that provides access to Microsoft file shares and support for common operations that allow shared files to be accessed or modified.

A CIFS browsing box is provided on the portal page to allow browsing and file access of files on the remote NBNS servers. NBNS servers are configured in the WebVPN context submode, and the NBNS list must be defined in the group policy for the given WebVPN context.

The following access capabilities can be configured for WebVPN:

- Network browse (listing of domains)
- Domain browse (listing of servers)
- Server browse (listing of shares)
- Listing files in a share
- Downloading files
- Modifying files

- Creating new directories
- Creating new files
- Deleting files

NBMS Resolution

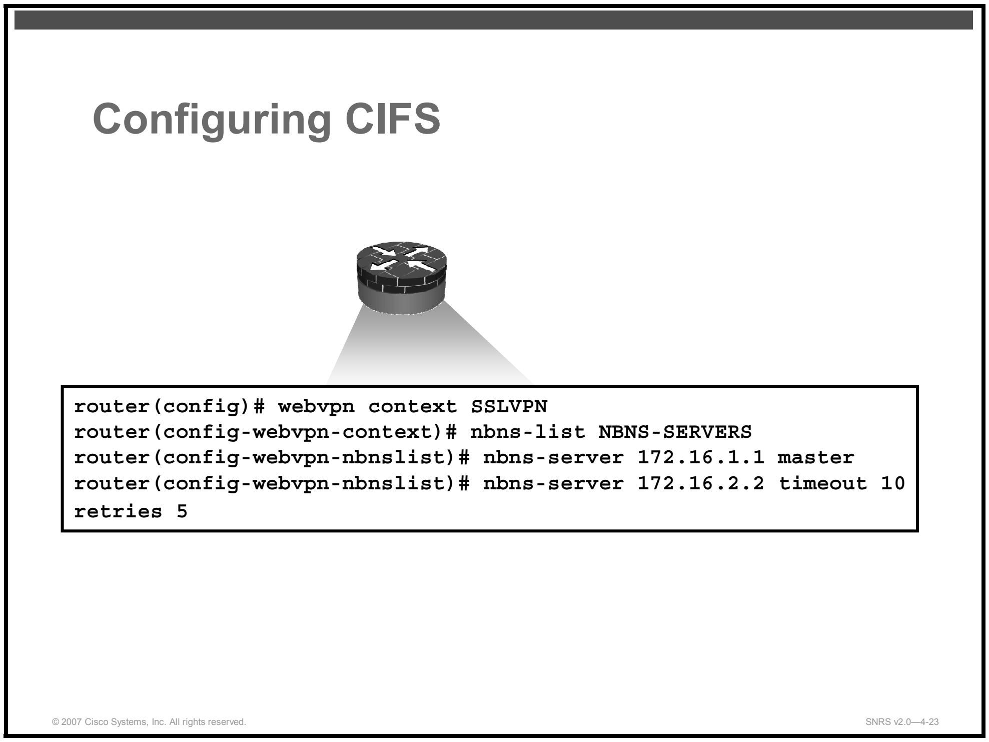
Microsoft Windows Internet Name Service (WINS) uses NBMS resolution to map and establish connections between Microsoft servers. A single server must be identified by its IP address in this configuration. Up to three servers can be added to the configuration. If multiple servers are added, one server should be configured as the master browser.

Samba Support

Microsoft file shares can be accessed through the browser on a Linux system that is configured to run Samba.

Restrictions

Only file shares configured on Microsoft Windows 2000 or Microsoft Windows XP servers are supported.



Configuring CIFS

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# nbns-list NBNS-SERVERS
router(config-webvpn-nbnslist)# nbns-server 172.16.1.1 master
router(config-webvpn-nbnslist)# nbns-server 172.16.2.2 timeout 10
retries 5
```

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-23

Configuring CIFS

Follow this procedure to enable file sharing support in WebVPN.

Step 1 Enter SSL VPN configuration mode.

```
router(config)# webvpn context SSLVPN
```

Step 7 Enter SSL VPN NBNS list configuration mode.

```
router(config-webvpn-context)# nbns-list name
```

Where:

- *Name:* This is the name of the NBNS list. The name can be up to 64 characters in length. This argument is case sensitive.

The NBNS server list is used to configure a list of Microsoft WINS to resolve Microsoft file directory shares. Entering the **nbns-list** command places the router in SSL VPN NBNS list configuration mode. You can specify up to three NBMS servers. A single server is configured as the master browser if multiple servers are specified in the server list.

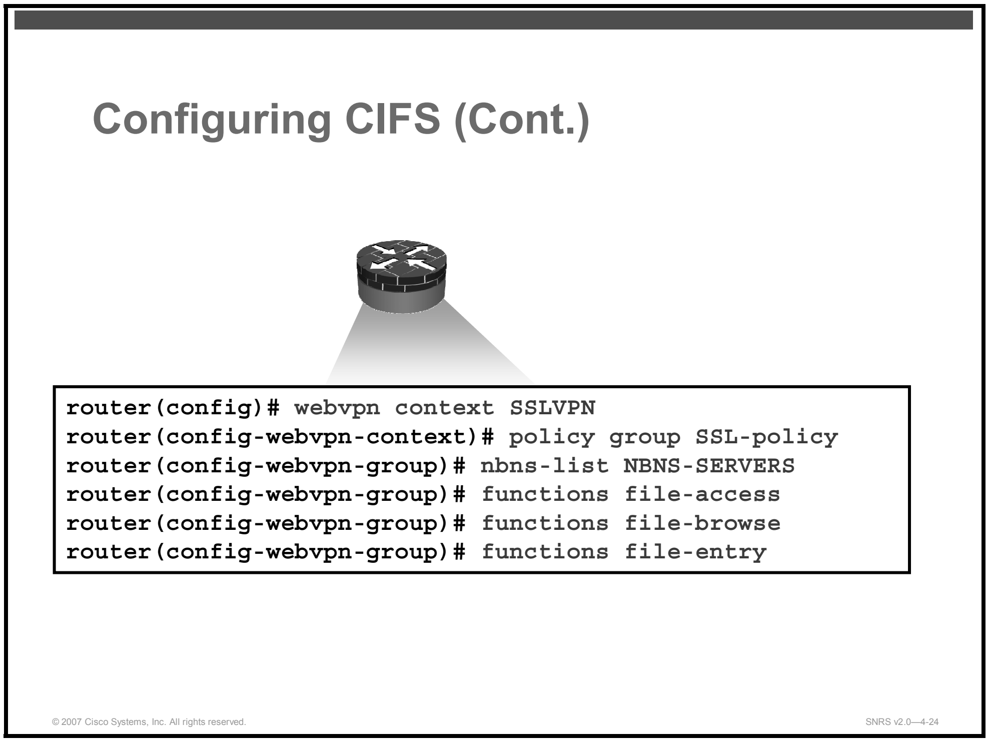
Step 2 Attach a NBNS server list to the policy group.

```
router(config-webvpn-nbnslist)# nbns-server ip-address
[master] [timeout seconds] [retries number]
```

Syntax Description

- *ip-address:* This is the IPv4 address of the NetBIOS server.
- **master:** (Optional) This configures a single NetBIOS server as the master browser.

- **timeout** *seconds:* (Optional) This configures the length of time, in seconds, that the networking device will wait for a query reply before sending a query to another NetBIOS server. A number from 1 through 30 can be configured for this argument. The default value is 2.
- **retries** *number:* (Optional) This is number of times that the specified NetBIOS server will be queried. A number from 0 through 10 can be configured for this argument. Entering the number 0 configures the networking device not to resend a query. The default value is 2.



Configuring CIFS (Cont.)

```
router(config)# webvpn context SSLVPN
router(config-webvpn-context)# policy group SSL-policy
router(config-webvpn-group)# nbns-list NBNS-SERVERS
router(config-webvpn-group)# functions file-access
router(config-webvpn-group)# functions file-browse
router(config-webvpn-group)# functions file-entry
```

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-24

After you finish configuring the NBNS parameters for the Microsoft WINS servers, you will need to attach the

Step 1 Enter SSL VPN configuration mode.

```
router(config)# webvpn context context-name
```

Step 2 Enter policy group configuration mode.

```
router(config-webvpn-context)# policy group group-name
```

Step 3 Attach an NBNS server list to a policy group.

```
router(config-webvpn-group)# nbns-list list-name
```

Syntax Description

name	Name of the NBNS server list configured in SSL VPN context configuration mode

Step 4 Configure access for Microsoft file shares.

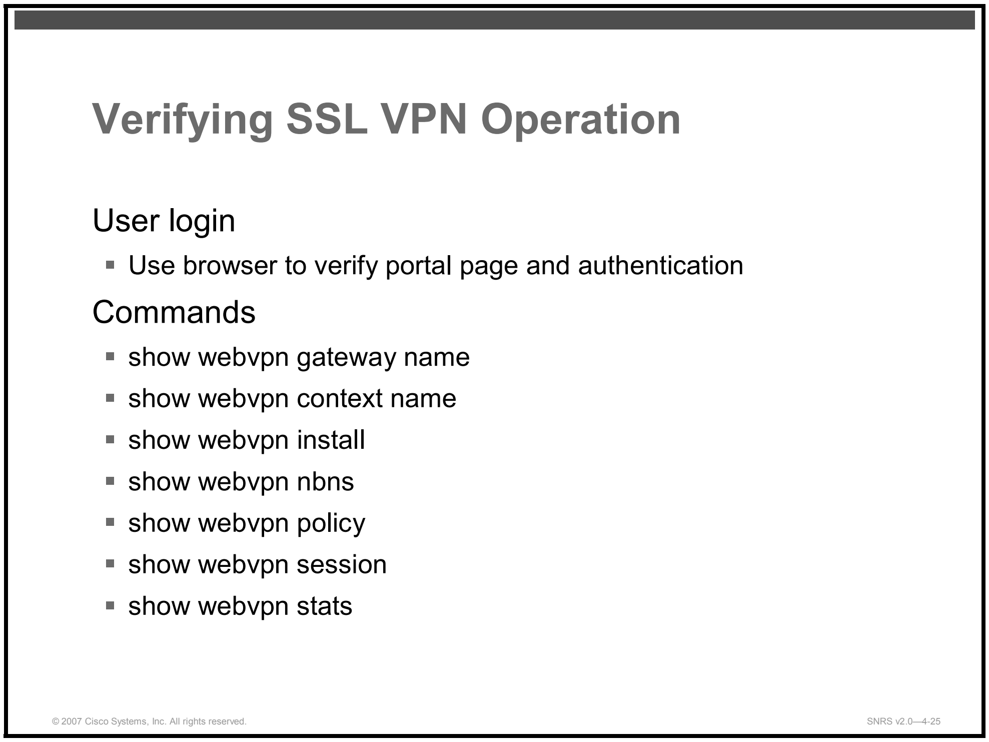
```
router(config-webvpn-group)# functions {file-access | file-browse |
file-entry | httpauth-disabled | svc-enabled | svc-required}
```

Syntax Description

`file-access`	This keyword enables network file share access. File servers in the server list are listed on the SSL VPN home page if this keyword is enabled.
`file-browse`	This keyword enables browse permissions for server and file shares. The file access function must be enabled to also use this function.
`file-entry`	This keyword enables “modify” permissions for files in the shares listed on the SSL VPN home page.
`httpauth-disabled`	This keyword disables Microsoft NT LAN Manager (NTLM) authentication. To reinstate NTLM authentication, use the no form of the functions command with the httpauth-disabled keyword.
svc-enabled	This keyword enables tunnel support for the user and allows the user of the group to use tunnel mode. If the Cisco SVC software package fails to install on the PC of the end user, the end user can continue to use clientless mode or thin-client mode.
svc-required	This keyword enables only tunnel support for the user. If the Cisco SVC software package fails to install on the PC of the end user, the other access modes cannot be used.

Verifying WebVPN Functionality

This topic describes commands used to monitor and maintain WebVPN.



Verifying SSL VPN Operation

User login

- Use browser to verify portal page and authentication

Commands

- show webvpn gateway name
- show webvpn context name
- show webvpn install
- show webvpn nbns
- show webvpn policy
- show webvpn session
- show webvpn stats

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-25

When verifying WebVPN operation and functionality, you can use a test login session and also some commands from the command-line interface (CLI).

User Login Verification

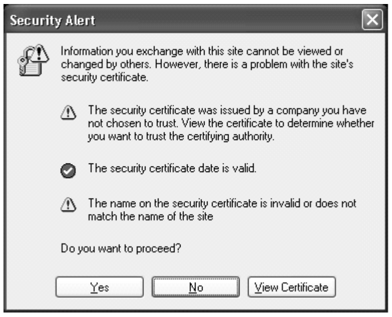
A remote user whose enterprise network has configured WebVPN can access the network by launching a browser and connecting to the WebVPN gateway. Remote users present their credentials to authenticates, and a portal page (home page) of the enterprise site is displayed. The portal page displays WebVPN features (for example, e-mail and web browsing) to which remote users have access on the basis of their credentials. If a remote user has access to all features enabled on the WebVPN gateway, the home page will provide access links.

This section describes the page flow process for a WebVPN session. When remote users enter the HTTP or HTTPS URL (http://address) into their browser, they are then redirected to https://address/index.html, where the login page is located.

Note Depending on the configuration of the browser, this redirection may cause a warning in the browser of the remote user indicating that the remote user is being redirected to a secure connection.

The following screens will show the user login process in action.

SSL/TLS Certificate


Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

⚠ The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

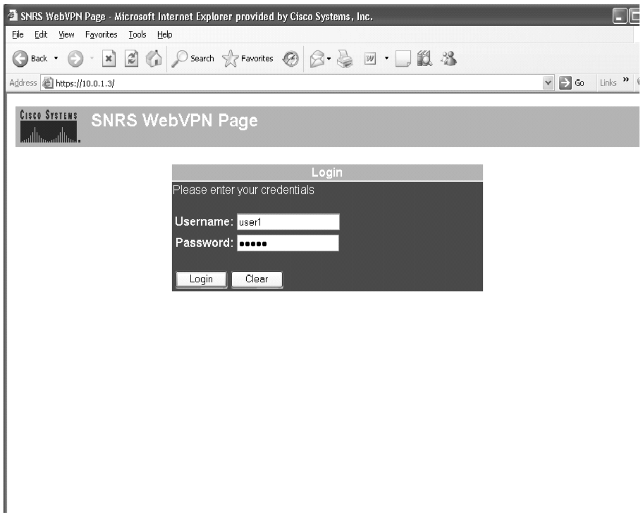
✓ The security certificate date is valid.

⚠ The name on the security certificate is invalid or does not match the name of the site

Do you want to proceed?

When the HTTPS connection is established, a warning about the SSL/Transport Layer Security (TLS) certificate may display. If the warning displays, the remote user should install this certificate. If the warning does not display, the system already has a certificate that the browser trusts. The remote user is then connected to the login page.

SSL VPN Login Page


SNRS WebVPN Page - Microsoft Internet Explorer provided by Cisco Systems, Inc.
File Edit View Favorites Tools Help
Back
Search
Favorites
Address https://10.0.1.3/
Go
Links
Cisco Systems
SNRS WebVPN Page
Login
Login
Clear

Please enter your credentials

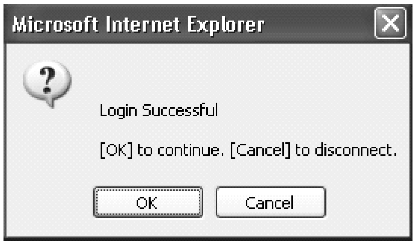
Username: user1
Password: •••••

SNRS v2.0—4-27

The login page prompts remote users to enter their username and password, which are entered into an HTML form. If an authentication failure occurs, the login page displays an error message.

The login page has logos, titles, messages, and colors that may be customized by administrators.

SSL VPN Login Successful

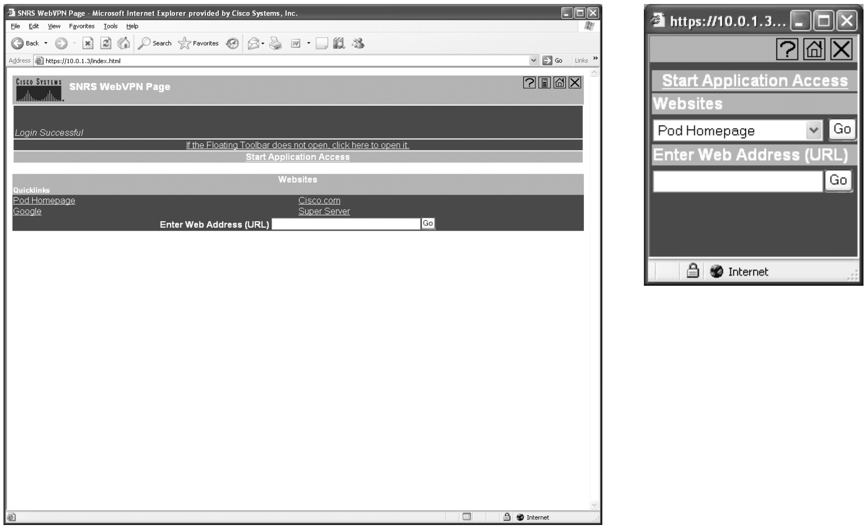


Login Successful

[OK] to continue. [Cancel] to disconnect.

This message will appear when a login has been successful.

SSL VPN Portal Page and Floating Toolbar



SNRS WebVPN Page - Microsoft Internet Explorer provided by Cisco Systems, Inc.

Address https://10.0.1.3/index.html

SNRS WebVPN Page

Login Successful

If the Floating Toolbar does not open, click here to open it.

Start Application Access

Websites

Quicklinks

Pod Homepage	Cisco.com
Google	Super Server

Enter Web Address (URL) Go

https://10.0.1.3...

Start Application Access

Websites

Pod Homepage Go

Enter Web Address (URL)

Go

The portal page is the main page for the WebVPN functionality. Items that you have not configured are not displayed on the portal page.

Note E-mail access is supported by thin-client mode, which is downloaded using the Start Application Access link.

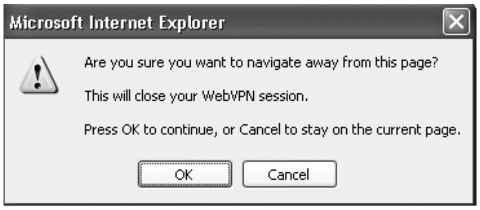
Remote Servers

Remote users may enter an address or URL path of a website that they want to visit either in the text box on the portal page or in the text box on the floating toolbar. Pages from the remote server are displayed in the browser window. The remote user can then browse to other links on the page. The figure illustrates the portal page of a typical website. By clicking the home icon button on the floating toolbar, the remote user can go back to the portal page.

WebVPN Floating Toolbar

A floating toolbar (see figure) allows the remote user to enter URLs, browse file locations, and choose preconfigured web connections without interfering with the main browser window. The floating toolbar represents the WebVPN session. The next figure shows what happens when the remote user clicks the window close button.

SSL VPN Logout Dialog



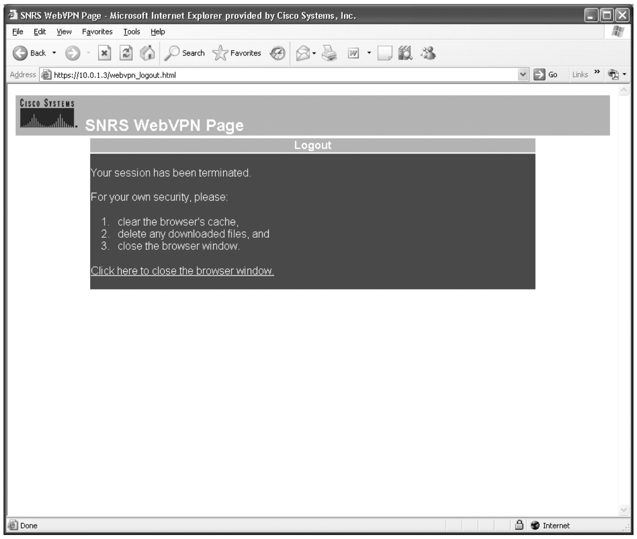
Are you sure you want to navigate away from this page?

This will close your WebVPN session.

Press OK to continue, or Cancel to stay on the current page.

If remote users click the window close button, the WebVPN gateway prompts them to confirm that they want to close the session.

SSL VPN Logout



SNRS WebVPN Page - Microsoft Internet Explorer provided by Cisco Systems, Inc.
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Cisco Systems
Done
Internet

https://10.0.1.3/webvpn_logout.html

SNRS WebVPN Page

Logout

Your session has been terminated.

For your own security, please:

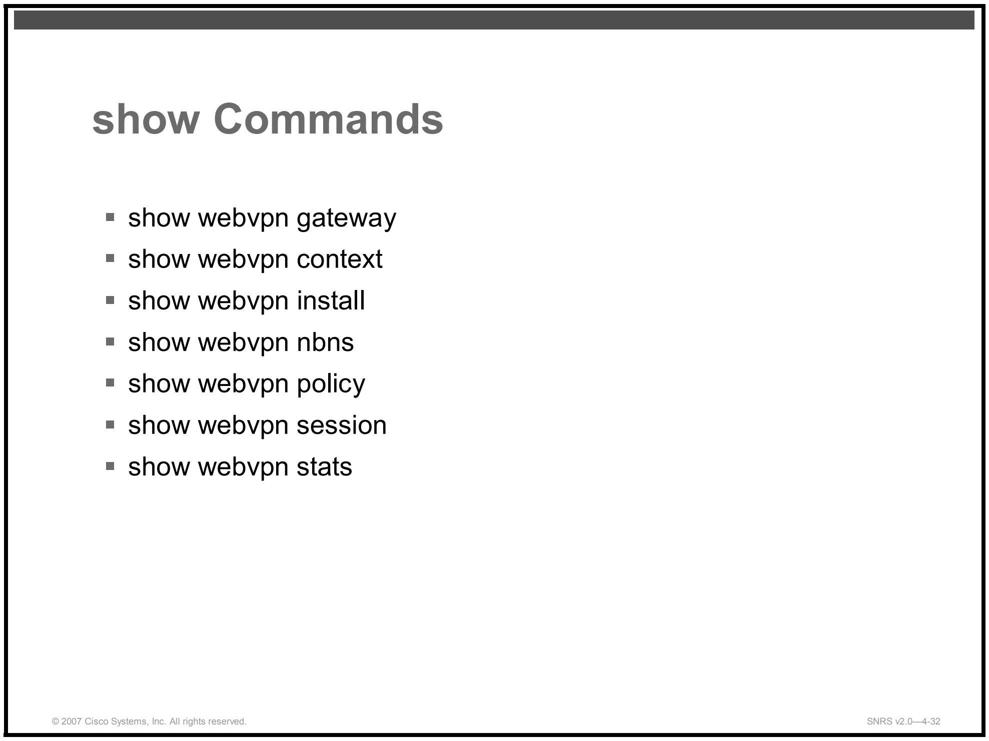
1. clear the browser's cache,
2. delete any downloaded files, and
3. close the browser window.

Click here to close the browser window.

The logout page displays if the remote user clicks the logout link, or if the session terminates because of an idle timeout or a maximum connection time.

Commands

This section covers some of the **show** commands available for viewing WebVPN information.



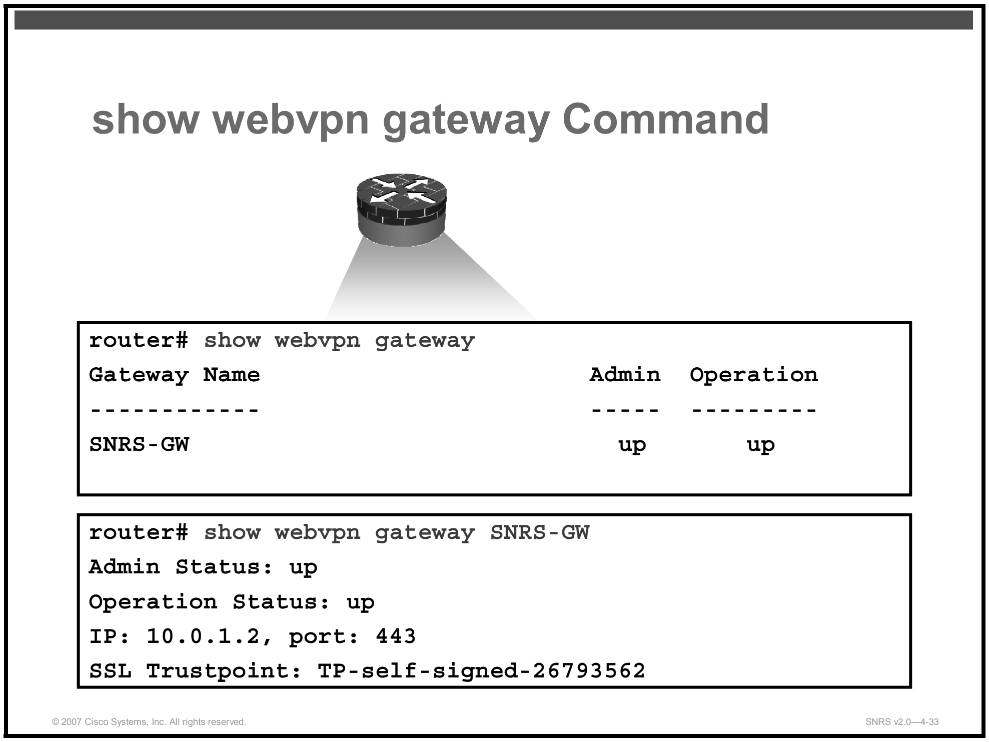
show Commands

- show webvpn gateway
- show webvpn context
- show webvpn install
- show webvpn nbns
- show webvpn policy
- show webvpn session
- show webvpn stats

SNRS v2.0—4-32

This section describes the **show** commands that are used to verify the following:

- **WebVPN gateway configuration:** Display the status of the WebVPN gateway using the **show webvpn gateway** command
- **WebVPN context configuration:** Display the operational status and configuration parameters for WebVPN context configurations using the **show webvpn context** command
- **Cisco Secure Desktop and Cisco SVC installation status:** Display the installation status of Cisco Secure Desktop and Cisco SVC client software packages using the **show webvpn install** command
- **NBNS information:** Display information in the NBNS cache using the **show webvpn nbns** command
- **WebVPN group policy configuration:** Display the context configuration associated with a policy group using the **show webvpn policy** command
- **WebVPN user session information:** Display WebVPN user session information using the **show webvpn session** command
- **WebVPN application statistics:** Display WebVPN application and network statistics using the **show webvpn stats** command

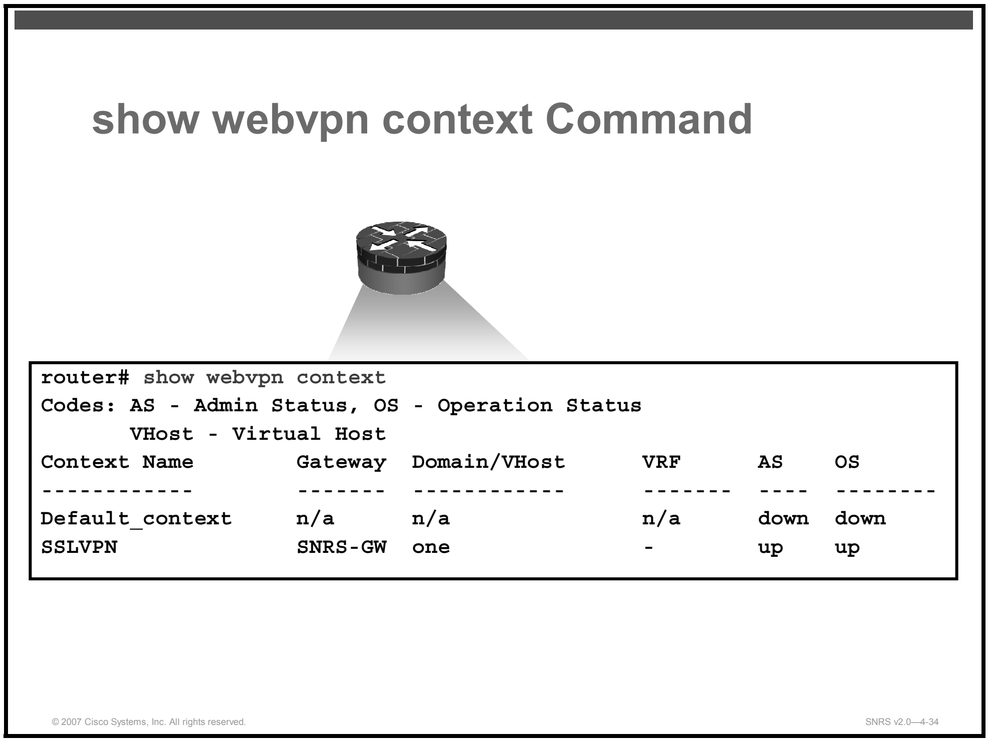


show webvpn gateway Command

```
router# show webvpn gateway
Gateway Name                          Admin  Operation
------------                          -----  ---------
SNRS-GW                                 up       up
```

```
router# show webvpn gateway SNRS-GW
Admin Status: up
Operation Status: up
IP: 10.0.1.2, port: 443
SSL Trustpoint: TP-self-signed-26793562
```

This figure is a sample output from the **show webvpn gateway** command, both alone and entered with a specific WebVPN gateway name.

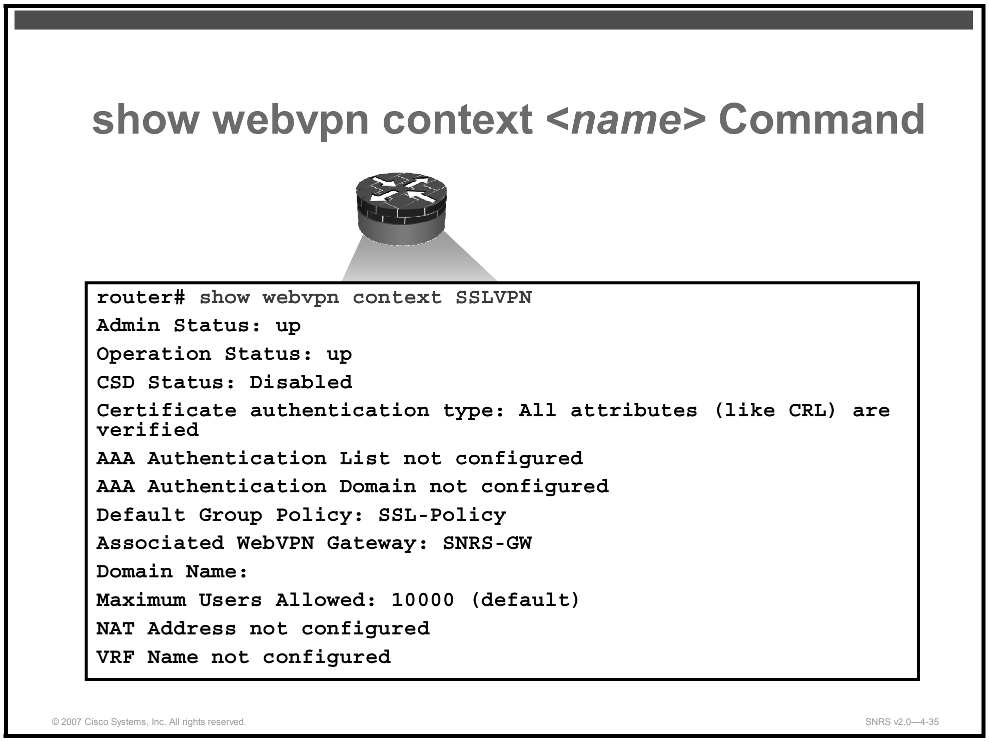


show webvpn context Command

```
router# show webvpn context
Codes: AS - Admin Status, OS - Operation Status
       VHost - Virtual Host
Context Name        Gateway  Domain/VHost      VRF      AS    OS
------------        -------  ------------      -------  ----  --------
Default_context     n/a      n/a               n/a      down  down
SSLVPN              SNRS-GW  one               -        up    up
```

SNRS v2.0—4-34

This figure is a sample output from the **show webvpn context** command.

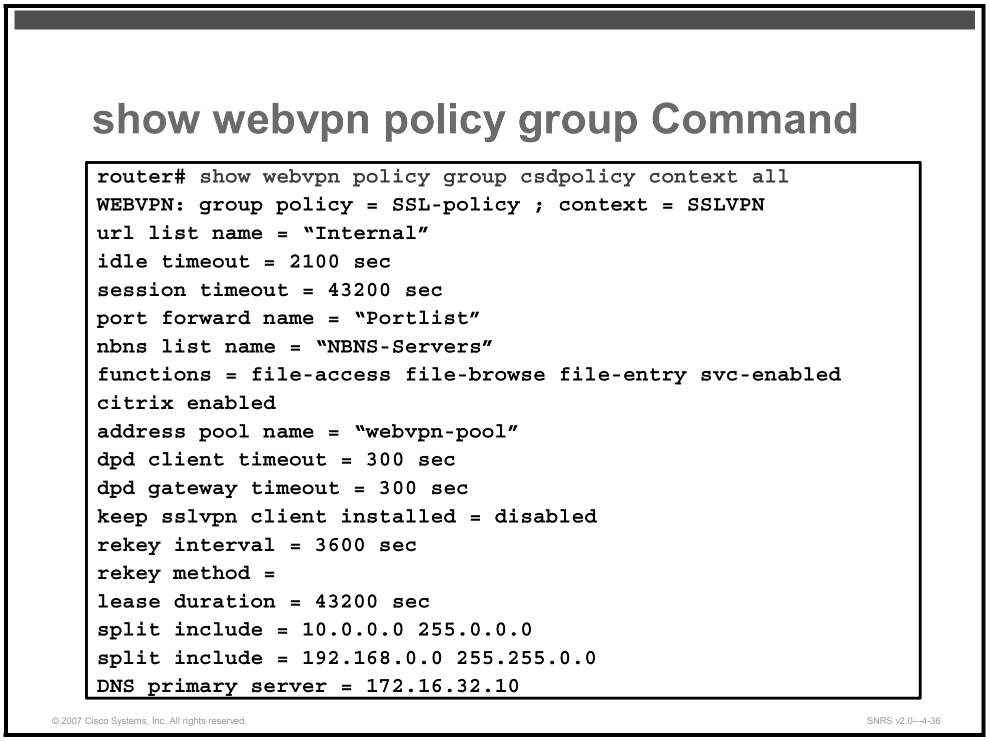


show webvpn context <*name*> Command

```
router# show webvpn context SSLVPN
Admin Status: up
Operation Status: up
CSD Status: Disabled
Certificate authentication type: All attributes (like CRL) are
verified
AAA Authentication List not configured
AAA Authentication Domain not configured
Default Group Policy: SSL-Policy
Associated WebVPN Gateway: SNRS-GW
Domain Name:
Maximum Users Allowed: 10000 (default)
NAT Address not configured
VRF Name not configured
```

SNRS v2.0—4-35

This figure is a sample output from the **show webvpn context <*name*>** command, entered with the name of a specific WebVPN context.

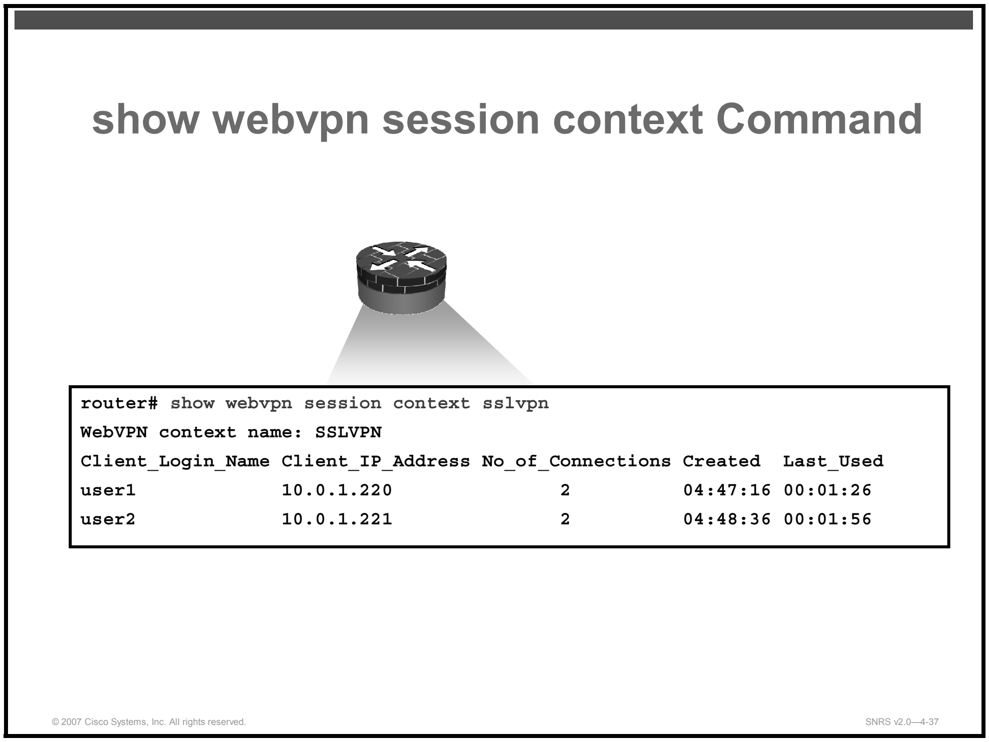


show webvpn policy group Command

```
router# show webvpn policy group csdpolicy context all
WEBVPN: group policy = SSL-policy ; context = SSLVPN
url list name = “Internal”
idle timeout = 2100 sec
session timeout = 43200 sec
port forward name = “Portlist”
nbns list name = “NBNS-Servers”
functions = file-access file-browse file-entry svc-enabled
citrix enabled
address pool name = “webvpn-pool”
dpd client timeout = 300 sec
dpd gateway timeout = 300 sec
keep sslvpn client installed = disabled
rekey interval = 3600 sec
rekey method =
lease duration = 43200 sec
split include = 10.0.0.0 255.0.0.0
split include = 192.168.0.0 255.255.0.0
DNS primary server = 172.16.32.10
```

SNRS v2.0—4-36

This figure is a sample output from the **show webvpn policy group** command.

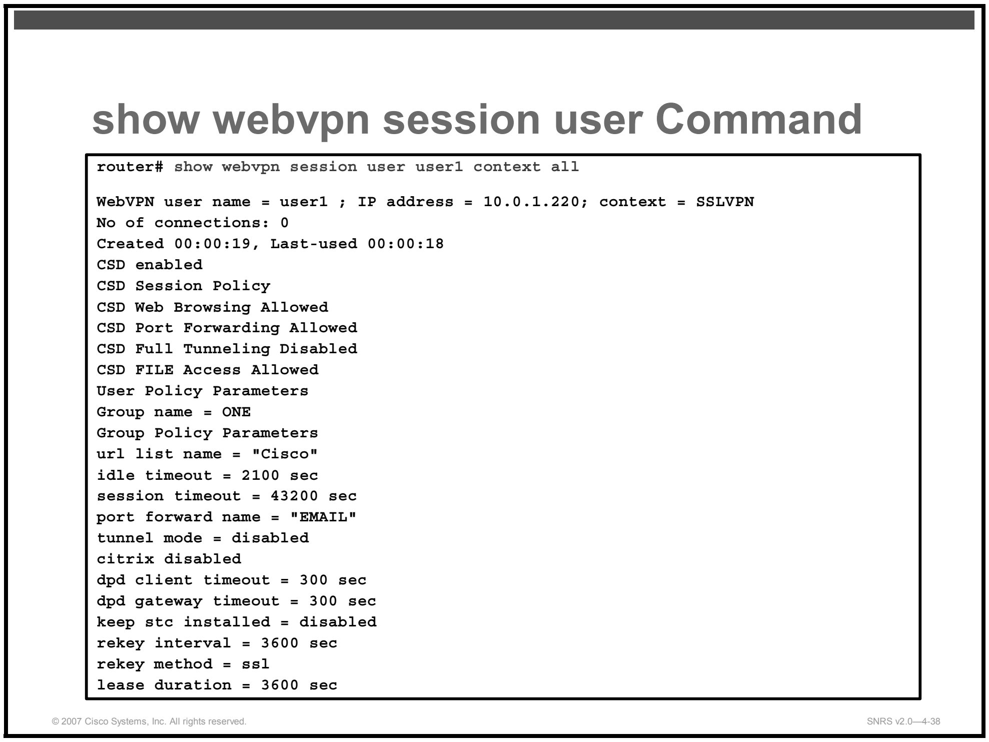


show webvpn session context Command

```
router# show webvpn session context sslvpn
WebVPN context name: SSLVPN
Client_Login_Name Client_IP_Address No_of_Connections Created  Last_Used
user1             10.0.1.220                2          04:47:16 00:01:26
user2             10.0.1.221                2          04:48:36 00:01:56
```

SNRS v2.0—4-37

This figure is a sample output from the **show webvpn session context** command.



show webvpn session user Command

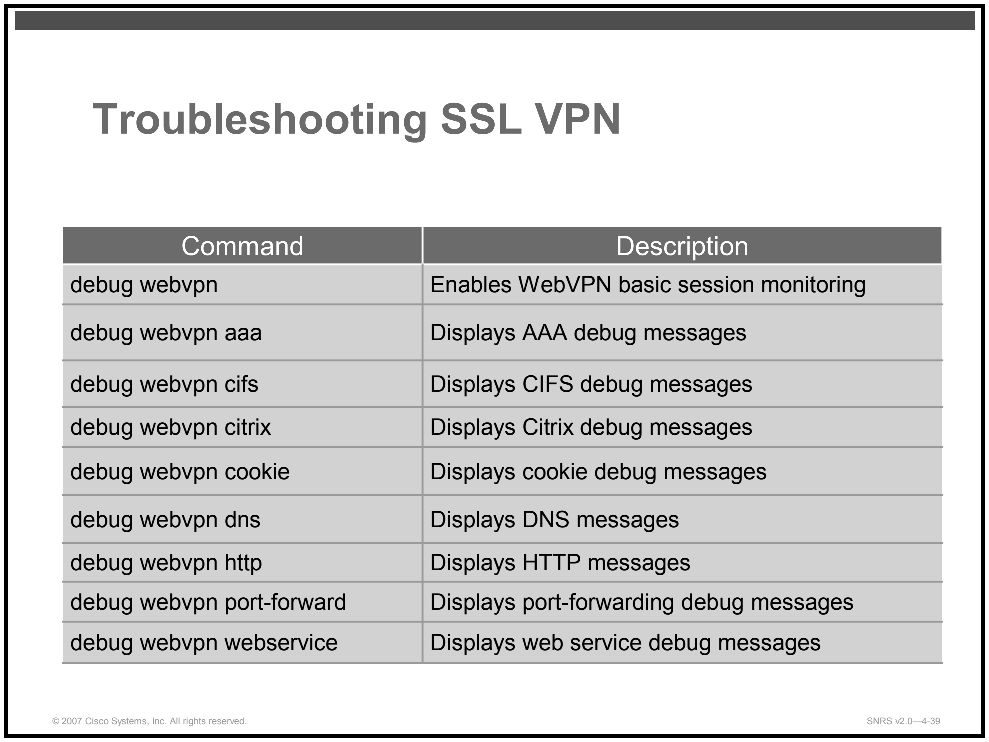
```
router# show webvpn session user user1 context all

WebVPN user name = user1 ; IP address = 10.0.1.220; context = SSLVPN
No of connections: 0
Created 00:00:19, Last-used 00:00:18
CSD enabled
CSD Session Policy
CSD Web Browsing Allowed
CSD Port Forwarding Allowed
CSD Full Tunneling Disabled
CSD FILE Access Allowed
User Policy Parameters
Group name = ONE
Group Policy Parameters
url list name = "Cisco"
idle timeout = 2100 sec
session timeout = 43200 sec
port forward name = "EMAIL"
tunnel mode = disabled
citrix disabled
dpd client timeout = 300 sec
dpd gateway timeout = 300 sec
keep stc installed = disabled
rekey interval = 3600 sec
rekey method = ssl
lease duration = 3600 sec
```

SNRS v2.0—4-38

This figure is a sample output from the **show webvpn session user** command.

Troubleshooting WebVPN

This topic lists the **debug** commands used to troubleshoot WebVPN.



Troubleshooting SSL VPN

Command	Description
debug webvpn	Enables WebVPN basic session monitoring
debug webvpn aaa	Displays AAA debug messages
debug webvpn cifs	Displays CIFS debug messages
debug webvpn citrix	Displays Citrix debug messages
debug webvpn cookie	Displays cookie debug messages
debug webvpn dns	Displays DNS messages
debug webvpn http	Displays HTTP messages
debug webvpn port-forward	Displays port-forwarding debug messages
debug webvpn webservice	Displays web service debug messages

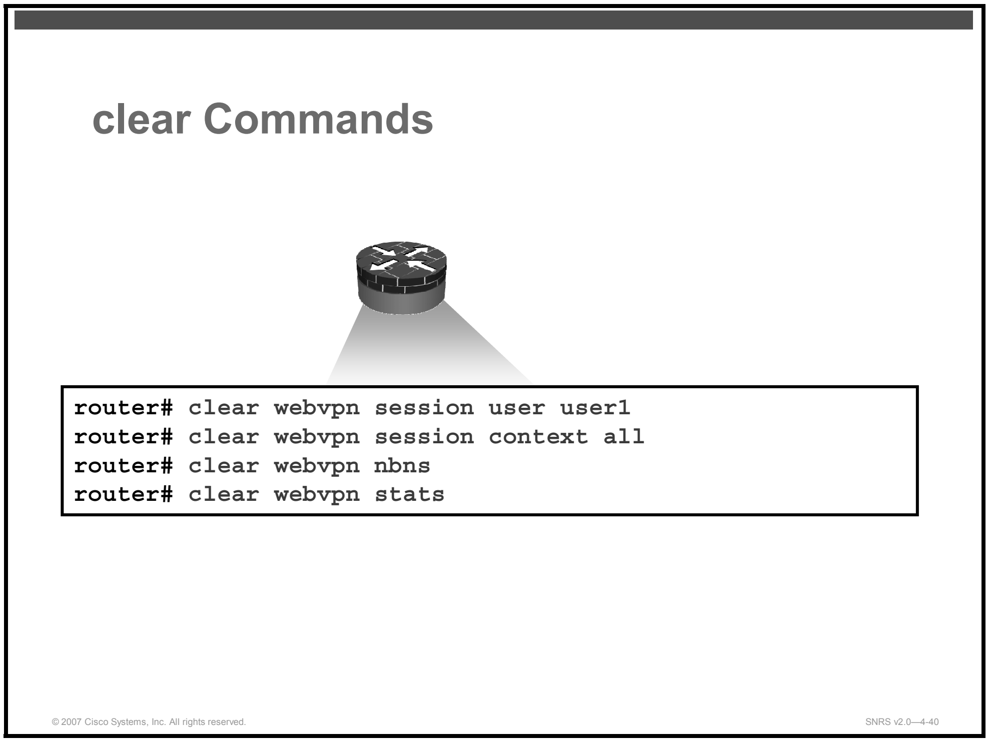
SNRS v2.0—4-39

There are many **debug webvpn** commands available to use in troubleshooting Cisco IOS SSL VPN.

The table shows some **debug** commands available to help troubleshooting possible problems.

Debug Commands

Command	Description
debug webvpn	Enables WebVPN basic session monitoring
debug webvpn aaa	Displays AAA debug messages
debug webvpn cifs	Displays CIFS debug messages
debug webvpn citrix	Displays Citrix debug messages
debug webvpn cookie	Displays cookie debug messages
debug webvpn dns	Displays DNS messages
debug webvpn http	Displays HTTP messages
debug webvpn port-forward	Displays port-forwarding debug messages
debug webvpn web service	Displays web service debug messages



clear Commands

```
router# clear webvpn session user user1
router# clear webvpn session context all
router# clear webvpn nbns
router# clear webvpn stats
```

WebVPN clear Commands

There are **clear** commands that will clear the NBNS cache, clear the WebVPN sessions, and clear the statistics.

To clear WebVPN remote user sessions, use the following command:

```
router# clear webvpn session {[user name] context {name |
all}}
```

Syntax Description

user *name*	(Optional) Clears session information for a specific user
context {*name* | **all**}	Clears session information for a specific context or all contexts

This command is used to clear the session for either the specified remote user or all remote users in the specified context.

To clear the NBNS cache on a WebVPN gateway, use the following command:

```
router# clear webvpn nbns [context {name | all}]
```

Syntax Description

context	(Optional) Clears NBNS statistics for a specific context or all contexts
name	Clears NBNS statistics for a specific context
all	Clears NBNS statistics for all contexts

Entering this command without any keywords or arguments clears all NBNS counters on the network device.

To clear (or reset) WebVPN application and access counters, use the following command:

```
router# clear webvpn stats [[cifs | citrix | mangle | port-
forward | tunnel] [context {name | all}]]
```

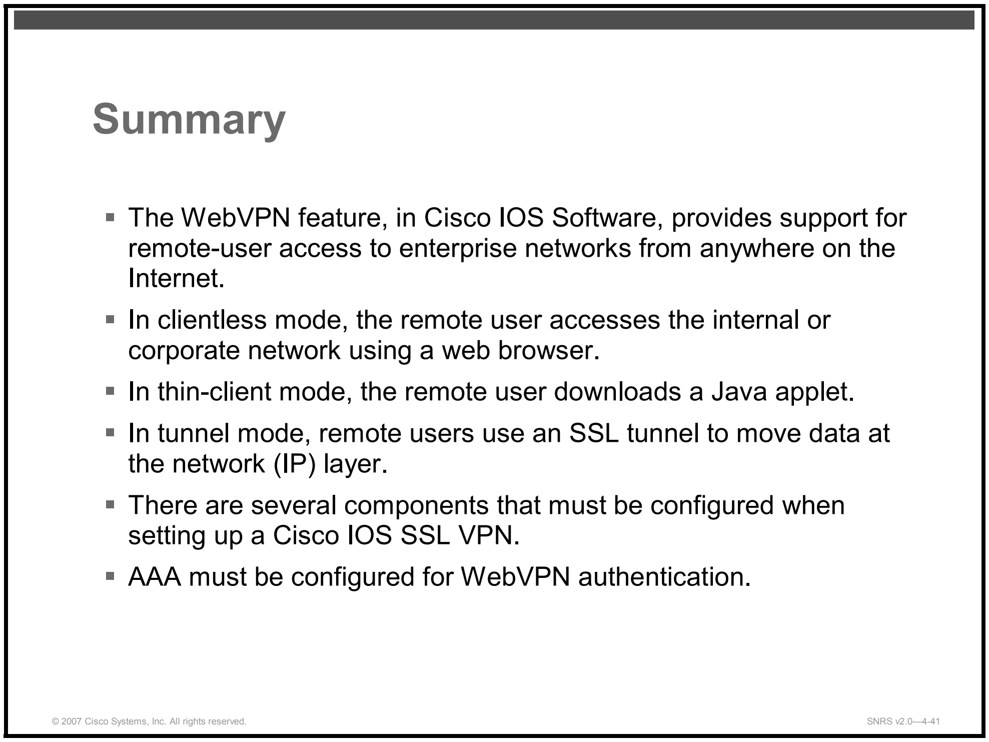
Syntax Description

cifs	(Optional) Clears Microsoft Windows file share (CIFS) statistics
citrix	(Optional) Clears Citrix application statistics
mangle	(Optional) Clears URL mangling statistics
port-forward	(Optional) Clears port-forwarding statistics
tunnel	(Optional) Clears Cisco SVC tunnel statistics
context {*name* | **all**}	(Optional) Clears information for either a specific context or all contexts

This command is used to clear counters for Microsoft Windows file shares, Citrix applications, URL mangling, application port forwarding, and Cisco SVC tunnels. The counters are cleared for either the specified context or all contexts on the WebVPN gateway.

Summary

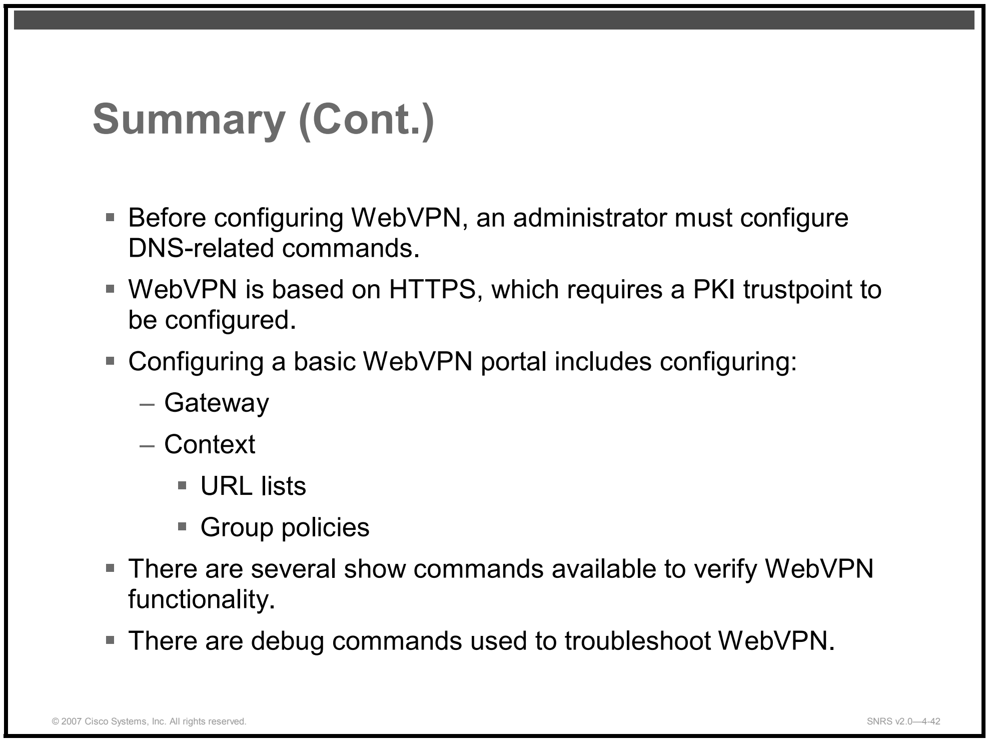
This topic summarizes the key points that were discussed in this lesson.



Summary

- The WebVPN feature, in Cisco IOS Software, provides support for remote-user access to enterprise networks from anywhere on the Internet.
- In clientless mode, the remote user accesses the internal or corporate network using a web browser.
- In thin-client mode, the remote user downloads a Java applet.
- In tunnel mode, remote users use an SSL tunnel to move data at the network (IP) layer.
- There are several components that must be configured when setting up a Cisco IOS SSL VPN.
- AAA must be configured for WebVPN authentication.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-41



Summary (Cont.)

- Before configuring WebVPN, an administrator must configure DNS-related commands.
- WebVPN is based on HTTPS, which requires a PKI trustpoint to be configured.
- Configuring a basic WebVPN portal includes configuring:
  - Gateway
  - Context
    - URL lists
    - Group policies
- There are several show commands available to verify WebVPN functionality.
- There are debug commands used to troubleshoot WebVPN.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-42

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Cisco IOS Software Releases 12.4T: SSL VPN (WebVPN).* http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a00805eeaea.html#wp1357310.
- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4: WebVPN.* http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_chapter09186a008044b201.html#wp1067202.
- Cisco Systems, Inc. *Enterprise Class Teleworker (ECT) Solution: SSL VPN Deployment Guide.* http://www.cisco.com/en/US/products/ps6660/products_white_paper0900aecd8029d630.shtml.

Lesson 8

Configuring Easy VPN Remote Access

Overview

Establishing a virtual private network (VPN) connection between two routers can be complicated and typically requires tedious coordination between network administrators to configure the VPN parameters of both routers. The Cisco Easy VPN Remote feature eliminates much of this work by implementing the Cisco VPN Client client/server protocol, which allows most VPN parameters to be defined at a Cisco Easy VPN Server. After the Cisco Easy VPN Server has been configured, a VPN connection can be created with minimal configuration on a Cisco Easy VPN Remote client. This lesson guides you through the configuration of Cisco Easy VPN Remote, including server and client operations.

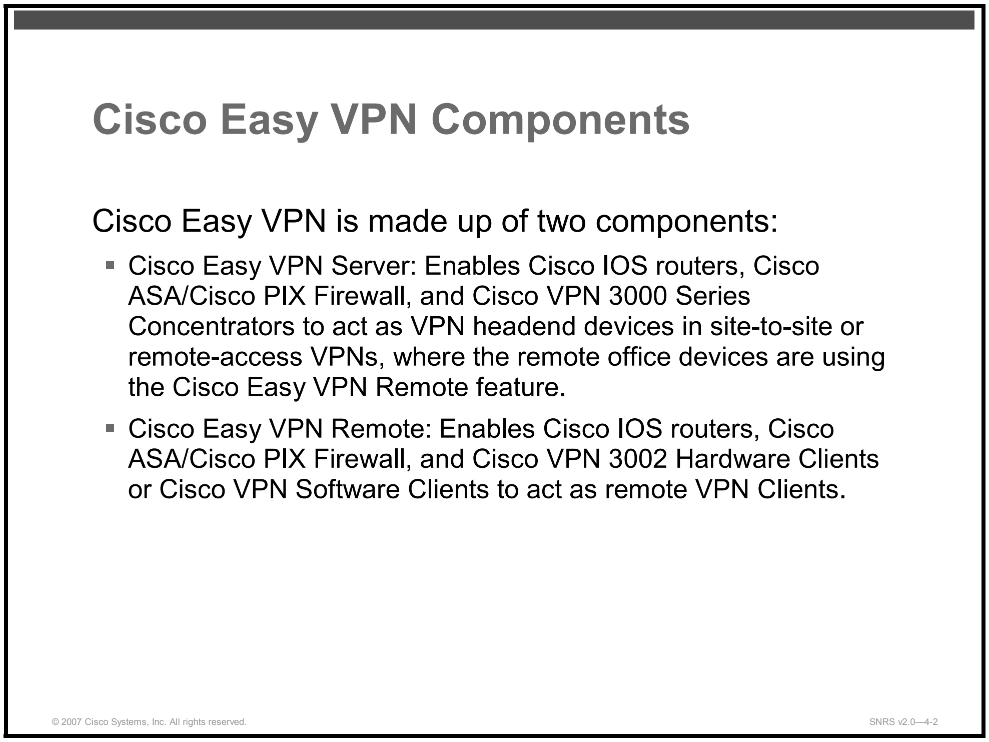
Objectives

Upon completing this lesson, you will be able to configure IP Security (IPsec) VPNs using Cisco Easy VPN. This ability includes being able to meet these objectives:

- Describe how Cisco Easy VPN provides Cisco IOS remote access
- Describe how to configure a router as a Cisco Easy VPN Remote access client
- Describe how to configure a Cisco Easy VPN Server to support Cisco Easy VPN Remote client access
- Configure the Cisco VPN Client v4.x

Introduction to Cisco Easy VPN

This topic describes how Cisco Easy VPN provides Cisco IOS remote access.



Cisco Easy VPN Components

Cisco Easy VPN is made up of two components:

- Cisco Easy VPN Server: Enables Cisco IOS routers, Cisco ASA/Cisco PIX Firewall, and Cisco VPN 3000 Series Concentrators to act as VPN headend devices in site-to-site or remote-access VPNs, where the remote office devices are using the Cisco Easy VPN Remote feature.
- Cisco Easy VPN Remote: Enables Cisco IOS routers, Cisco ASA/Cisco PIX Firewall, and Cisco VPN 3002 Hardware Clients or Cisco VPN Software Clients to act as remote VPN Clients.

SNRS v2.0—4-2

Cable modems, xDSL routers, and other forms of broadband access provide high-performance connections to the Internet; however, many applications also require the security of VPN connections that perform a high level of authentication and that encrypt the data between two particular endpoints.

Cisco Easy VPN simplifies the configuration of VPNs using routers as Easy VPN servers and clients.

A Cisco Easy VPN server can be a dedicated VPN device, such as a Cisco VPN 3000 Series Concentrator, a Cisco ASA, Cisco PIX Firewall, or a Cisco IOS router that supports the Cisco VPN Client client/server protocol.

After the Cisco Easy VPN Server has been configured, a VPN connection can be created with minimal configuration on a Cisco Easy VPN remote router, such as a Cisco 800 Series Router. When the Cisco Easy VPN remote router initiates the VPN tunnel connection, the Cisco Easy VPN Server pushes the IPsec policies to the Cisco Easy VPN remote router and creates the corresponding VPN tunnel connection.

The Cisco Easy VPN Remote feature provides for automatic management of the following details:

- Negotiating tunnel parameters, such as addresses, algorithms, and lifetime
- Establishing tunnels according to the parameters that were set
- Automatically creating the Network Address Translation (NAT) or Port Address Translation (PAT) and associated access control lists (ACLs) that are needed, if any
- Authenticating users—that is, ensuring that users are who they say they are—by usernames, group names, and passwords
- Managing security keys for encryption and decryption
- Authenticating, encrypting, and decrypting data through the tunnel

Cisco Easy VPN consists of two components: Cisco Easy VPN Server and Cisco Easy VPN Remote.

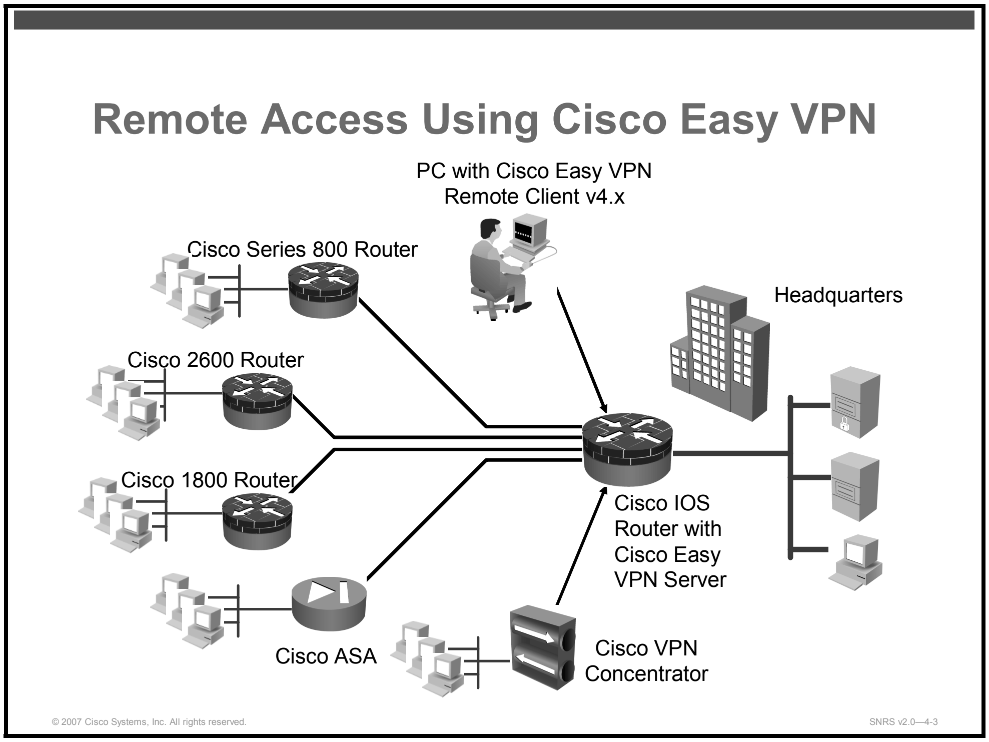
Cisco Easy VPN Server

Cisco Easy VPN Server enables Cisco IOS routers, Cisco ASA and Cisco PIX Firewalls, and Cisco VPN 3000 Series Concentrators to act as VPN headend devices in site-to-site or remote-access VPNs, where the remote office devices are using the Cisco Easy VPN Remote feature. Using this feature, security policies defined at the headend are pushed to the remote VPN device, ensuring that those connections have up-to-date policies in place before the connection is established.

In addition, a Cisco Easy VPN Server-enabled device can terminate IPsec tunnels initiated by mobile remote workers running VPN client software on PCs. This flexibility makes it possible for mobile and remote workers, such as salespeople on the road or telecommuters, to access their headquarters intranet, where critical data and applications exist.

Cisco Easy VPN Remote

Cisco Easy VPN Remote enables Cisco IOS routers, Cisco PIX Firewalls, and Cisco VPN 3002 Hardware Clients or Cisco VPN Software Clients to act as Cisco Easy VPN Remote clients. These devices can receive security policies from a Cisco Easy VPN Server, minimizing VPN configuration requirements at the remote location. This cost-effective solution is ideal for remote offices with little IT support or for large customer premises equipment (CPE) deployments where it is impractical to individually configure multiple remote devices. This feature makes VPN configuration as easy as entering a password. This, in turn, increases productivity and lowers costs because the need for local IT support is minimized.



Remote Access Using Cisco Easy VPN

SNRS v2.0—4-3

In the example in the figure, the VPN gateway is a Cisco IOS router running the Cisco Easy VPN Server feature. Remote Cisco IOS routers and Cisco VPN Software Clients connect to the Cisco Easy VPN Server for access to the corporate intranet.

Restrictions for Cisco Easy VPN Remote

There are a number of restrictions associated with Cisco Easy VPN Remote.

Required Cisco Easy VPN Servers

The Cisco Easy VPN Remote feature requires that the destination peer be a Cisco Easy VPN Server or Cisco VPN concentrator that supports the Cisco Easy VPN Server feature. At the time of publication, this includes the following platforms when running the indicated software releases:

- **Cisco 806 Broadband Router, Cisco 826 ADSL Router, Cisco 827 ADSL Router, Cisco 828 ADSL Router, Cisco 831** Ethernet Broadband Router**, Cisco 836** ADSL over ISDN Broadband Router**, and Cisco 837** ADSL Broadband Router**:** Cisco IOS Release 12.2(8)T or later. Cisco 800 Series Routers are not supported in Cisco IOS Release 12.3(7)XR, but they are supported in Cisco IOS Release 12.3(7)XR2.
- **Cisco 1700 Series Modular Access Routers:** Cisco IOS Release 12.2(8)T or later
- **Cisco 1800 Series Fixed Configuration Routers**: Cisco IOS Release 12.3(8)YI
- **Cisco 1812 Integrated Services Router:** Cisco IOS Release 12.3(8)YH
- **Cisco 2600 Series Multiservice Platforms:** Cisco IOS Release 12.2(8)T or later
- **Cisco 3620, 3640, 3660 Multiservice Platform:** Cisco IOS Release 12.2(8)T or later
- **Cisco 7100 Series VPN Routers:** Cisco IOS Release 12.2(8)T or later release
- **Cisco 7200 Series Routers:** Cisco IOS Release 12.2(8)T or later release
- **Cisco 7500 Series Routers:** Cisco IOS Release 12.2(8)T or later release

- **Cisco PIX 500 Series Security Appliances:** Cisco IOS Release 6.2 or later release
- **Cisco VPN 3000 Series Concentrators:** Cisco IOS Release 3.11 or later release

Only ISAKMP Group 2 Supported on Cisco Easy VPN Servers

The Cisco VPN Client client/server protocol supports only Internet Security Association and Key Management Protocol (ISAKMP) policies that use Diffie-Hellman (DH) group 2 (1024-bit DH) Internet Key Exchange (IKE) negotiation; therefore, the Cisco Easy VPN Server that is being used with the Cisco Easy VPN Remote feature must be configured for a group 2 ISAKMP policy. The Cisco Easy VPN Server cannot be configured for ISAKMP group 1 or group 5 when being used with a Cisco Easy VPN client.

Transform Sets Supported

To ensure a secure tunnel connection, the Cisco Easy VPN Remote feature does not support transform sets that provide encryption without authentication (**esp-des** and **esp-3des**) or transform sets that provide authentication without encryption (**esp-null esp-sha-hmac** and **esp-null esp-md5-hmac**).

Note The Cisco VPN Client client/server protocol does not support Authentication Header (AH) authentication, but Encapsulating Security Protocol (ESP) is supported.

Dial Backup for Cisco Easy VPN Remote Clients

Line status-based backup is not supported in this feature.

NAT Interoperability Support

NAT interoperability is not supported in client mode with split tunneling.

Virtual IPsec Interface Restrictions

Here are some of the restrictions of the Virtual IPsec Support feature:

- For the Virtual IPsec Interface Support feature to work, virtual templates support is needed.
- If you are using a virtual tunnel interface on the Cisco Easy VPN Remote device, it is recommended that you configure the server for a virtual tunnel interface.

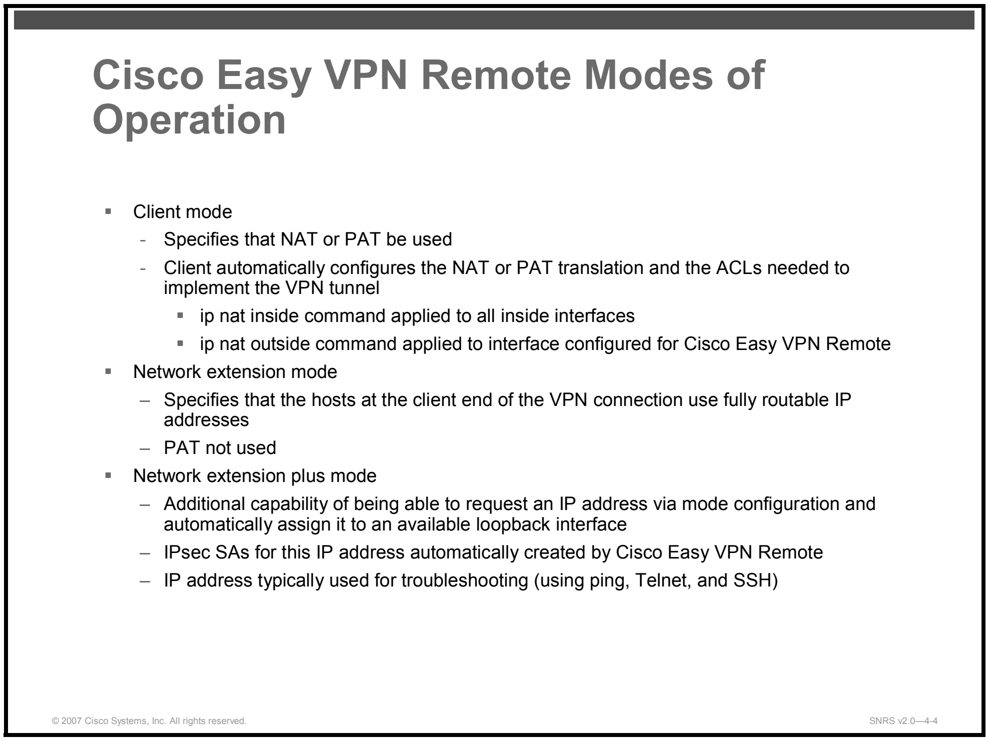
Dual Tunnel Support

The following restrictions apply if you are using dual tunnels that share common inside and outside interfaces:

- If dual tunnels are configured, one of the tunnels should have a split tunnel configured on the server.
- Web intercept can be configured for only one of the tunnels. Web intercept should not be used for the voice tunnel.
- Web intercept cannot be used for IP phones until the authorization proxy becomes aware of how to bypass the IP phone.
- Some features, such as Pushing a Configuration URL Through a Mode-Configuration Exchange, can be used only through a single tunnel.

Cisco Easy VPN Remote Modes of Operation

This topic describes the three modes of Cisco Easy VPN that are available for configuration on your router.



Cisco Easy VPN Remote Modes of Operation

- Client mode
  - Specifies that NAT or PAT be used
  - Client automatically configures the NAT or PAT translation and the ACLs needed to implement the VPN tunnel
    - ip nat inside command applied to all inside interfaces
    - ip nat outside command applied to interface configured for Cisco Easy VPN Remote
- Network extension mode
  - Specifies that the hosts at the client end of the VPN connection use fully routable IP addresses
  - PAT not used
- Network extension plus mode
  - Additional capability of being able to request an IP address via mode configuration and automatically assign it to an available loopback interface
  - IPsec SAs for this IP address automatically created by Cisco Easy VPN Remote
  - IP address typically used for troubleshooting (using ping, Telnet, and SSH)

The Cisco Easy VPN Remote feature supports three modes of operation: client, network extension, and network extension plus.

- **Client mode:** This mode specifies that NAT or PAT be configured to allow PCs and hosts on the client side of the VPN connection to form a private network that does not use any IP addresses in the IP address space of the destination server. Client mode automatically configures the NAT or PAT translation and ACLs that are needed to implement the VPN connection. These configurations are automatically (but temporarily) created when the VPN connection is initiated. When the tunnel is torn down, the NAT or PAT and ACL configurations are automatically deleted.

An enhancement has been made so that the IP address that is received via mode configuration is automatically assigned to an available loopback interface. The IPsec security associations (SAs) for this IP address are automatically created by Cisco Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell [SSH]).

The NAT or PAT configuration is created with the following assumptions:

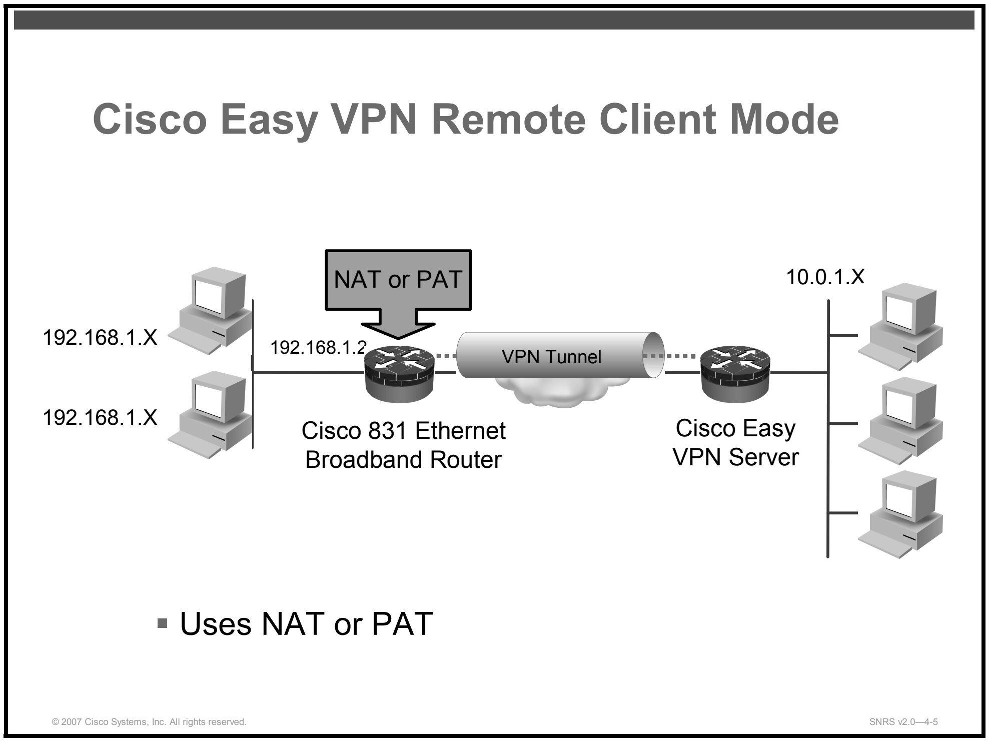
- The **ip nat inside** command is applied to all inside interfaces, including default inside interfaces. The default inside interface is Ethernet0 for the Cisco 800 Series Routers and the Cisco uBR900 Series Cable Access Routers. The default inside interface is FastEthernet0 for Cisco 1700 Series Modular Access Routers.

  - — The **ip nat outside** command is applied to the interface that is configured for Cisco Easy VPN Remote. On the Cisco uBR905 Cable Access Router and the Cisco uBR925 Cable Access Router, this is always the cable-modem0 interface. On the Cisco 800 Series Routers and Cisco 1700 Series Modular Access Routers, this is the outside interface configured for Cisco Easy VPN Remote. The Cisco 1700 Series Modular Access Routers can have multiple outside interfaces configured.

- **Network extension mode:** This mode specifies that the PCs and other hosts at the client end of the VPN tunnel should be given IP addresses that are fully routable and reachable by the destination network over the tunneled network so that they form one logical network. PAT is not used, which allows the client PCs and hosts to have direct access to the PCs and hosts at the destination network.

- **Network extension plus mode:** This mode is identical to network extension mode, with the additional capability of being able to request an IP address via mode configuration and automatically assign it to an available loopback interface. The IPsec SAs for this IP address are automatically created by Cisco Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and SSH).

All modes of operation also optionally support split tunneling, which allows secure access to corporate resources through the VPN tunnel while also allowing Internet access through a connection to an Internet service provider (ISP) or other service, eliminating the corporate network from the path for web access.



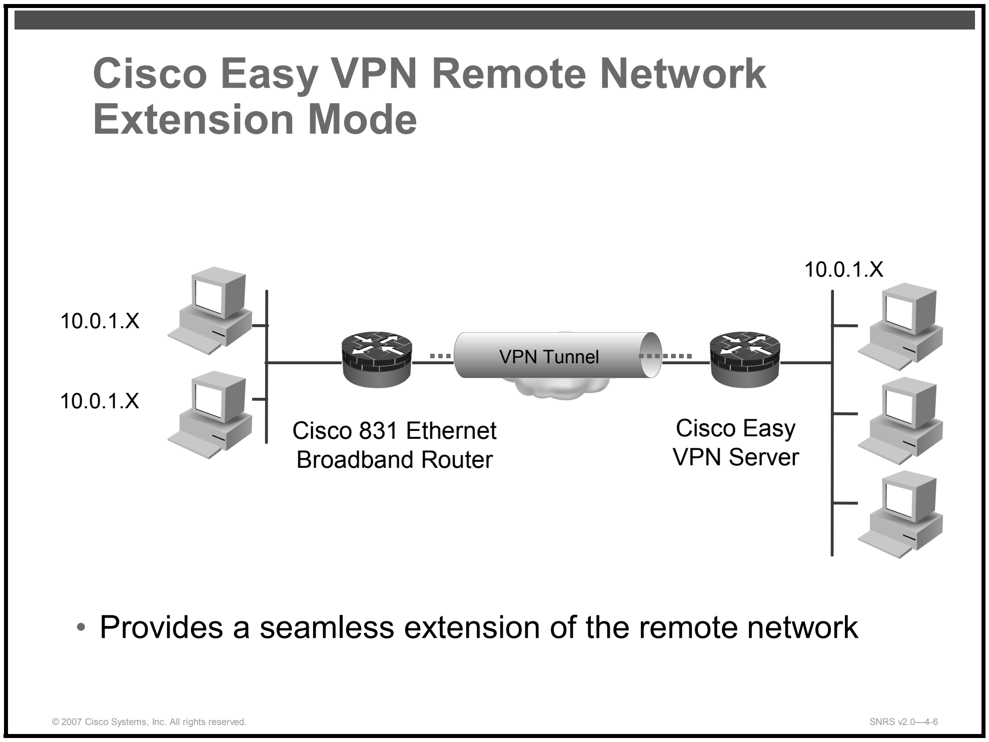
Cisco Easy VPN Remote Client Mode

- Uses NAT or PAT

SNRS v2.0—4-5

The figure illustrates the Cisco Easy VPN Remote client mode of operation. In this example, the Cisco 831 Ethernet Broadband Router provides access to two PCs, which have IP addresses in the 192.168.1.0 private network space. These PCs connect to the Ethernet interface on the Cisco 831 Ethernet Broadband Router, which also has an IP address in the 192.168.1.0 private network space. The Cisco 831 Ethernet Broadband Router performs NAT or PAT translation over the VPN tunnel so that the PCs can access the destination network.

Note	The diagram could also represent a split tunneling connection, in which the client PCs can access public resources in the global Internet without including the corporate network in the path for the public resources.



Cisco Easy VPN Remote Network Extension Mode

- Provides a seamless extension of the remote network

SNRS v2.0—4-6

This figure illustrates the network extension mode of operation. In this example, the Cisco 831 Ethernet Broadband Router acts as a Cisco Easy VPN Remote device, connecting to a router used as a Cisco Easy VPN Server.

The client hosts are given IP addresses that are fully routable by the destination network over the tunnel. These IP addresses could be either in the same subnet space as the destination network or in separate subnets, assuming that the destination routers are configured to properly route those IP addresses over the tunnel.

In this example, the PCs and hosts attached to the two routers have IP addresses that are in the same address space as the destination enterprise network. The PCs connect to the Ethernet interface of the Cisco 831 Ethernet Broadband Router, which also has an IP address in the enterprise address space.

This scenario provides a seamless extension of the remote network.

Authentication

The Cisco Easy VPN Remote feature supports a two-stage process for authenticating the remote router to the central Cisco VPN concentrator. The first step is group-level authentication and is part of the control channel creation. In this first stage, two types of authentication credentials can be used: either pre-shared keys or digital certificates. This discussion provides details about these options.

The second authentication step is called Extended Authentication (XAUTH). In this step, the remote side (in this case the Cisco Easy VPN router) submits a username and password to the central site router. This step is the same process as that which occurs when users of the Cisco VPN Software Client on a PC enter their username and password to activate their VPN tunnel. When using the router, the difference is that the router itself is being authenticated to the network, not a PC with Cisco VPN Software Client. XAUTH is an optional step (it can be disabled) but is normally enabled to improve security. After XAUTH is successful and the tunnel comes up, all PCs behind the Cisco Easy VPN Remote router have access to the tunnel.

If XAUTH is enabled, it is essential to decide how to input the username and password. There are two options. The first option is to store the XAUTH username and password in the configuration file of the router. This option is typically used if the router is shared between several PCs and the goal is to keep the VPN tunnel up all the time or to have the router automatically bring up the tunnel whenever there is data to be sent. An example of this application is a branch office situation, in which the users in the branch office want the VPN tunnel to be available whenever they have data to send and do not want to have to do anything special to activate the VPN tunnel. If the PCs in the branch office must be individually authenticated on the basis of the ID of each user, the correct configuration is to put the Cisco Easy VPN router in automatic activation mode to keep the tunnel “up” all the time and to use Cisco IOS Authentication Proxy (auth-proxy) or 802.1x to authenticate the individual PCs. Because the tunnel is always up, auth-proxy or 802.1x can access a central-site user database such as authentication, authorization, and accounting (AAA) or RADIUS to authenticate the individual user requests as they are submitted by PC users.

The second option for entry of the XAUTH username and password is not to store it on the router. Instead, a PC user who is connected to the router is presented with a special web page that allows the user to manually enter the username and password. The router sends the username and password to the central site Cisco VPN concentrator, and if the username and password are correct, the tunnel comes up. The typical application for this configuration is a teleworker network. Teleworkers want to control when the tunnel is up and have to enter their personal user credentials (which could include one-time passwords [OTPs]) to activate the tunnel. Also, the network administrator may want teleworker tunnels up only when someone is using them to conserve resources on the central Cisco VPN concentrators.

The XAUTH username and password can also be manually entered from the command-line interface (CLI) of the router. This method is not recommended for most situations because the user must first log in to the router (and needs a user ID on the router to do so). However, this method can be useful for network administrators during troubleshooting.

Using Pre-Shared Keys

Using pre-shared keys, each peer is aware of the key of the other peer. Pre-shared keys are displayed in running configurations, so they can be seen by anyone (referred to as clear text). When a more secure type of authentication is required, Cisco IOS Software also supports another type of pre-shared key—the encrypted pre-shared key.

Using an encrypted pre-shared key for authentication allows you to securely store plaintext passwords in type 6 (encrypted) format in NVRAM. A group pre-shared key can be preconfigured on both VPN tunnel peers. The encrypted form of the keyword can be seen in the running configuration, but the actual keyword is not visible.

Using Digital Certificates

Digital certificates provide for the support of Rivest, Shamir, and Adelman (RSA) signatures on Cisco Easy VPN Remote devices. The support is provided through an RSA certificate that can be stored on or off the remote device.

Note The recommended timeout for Cisco Easy VPN using digital certificates is 40 seconds.

Using XAUTH

XAUTH is an additional level of authentication that can be used. XAUTH is applicable when either group pre-shared keys or digital certificates are used. XAUTH credentials can be entered using a web interface manager, such as Cisco Router and Security Device Manager (SDM), or using the CLI.

The Save Password feature allows the XAUTH username and password to be saved in the Cisco Easy VPN Remote configuration so that you are not required to enter the username and password manually. OTPs are not supported by the Save Password feature and must be entered manually when XAUTH is requested. The Cisco Easy VPN Server must be configured to “Allow Saved Passwords.”

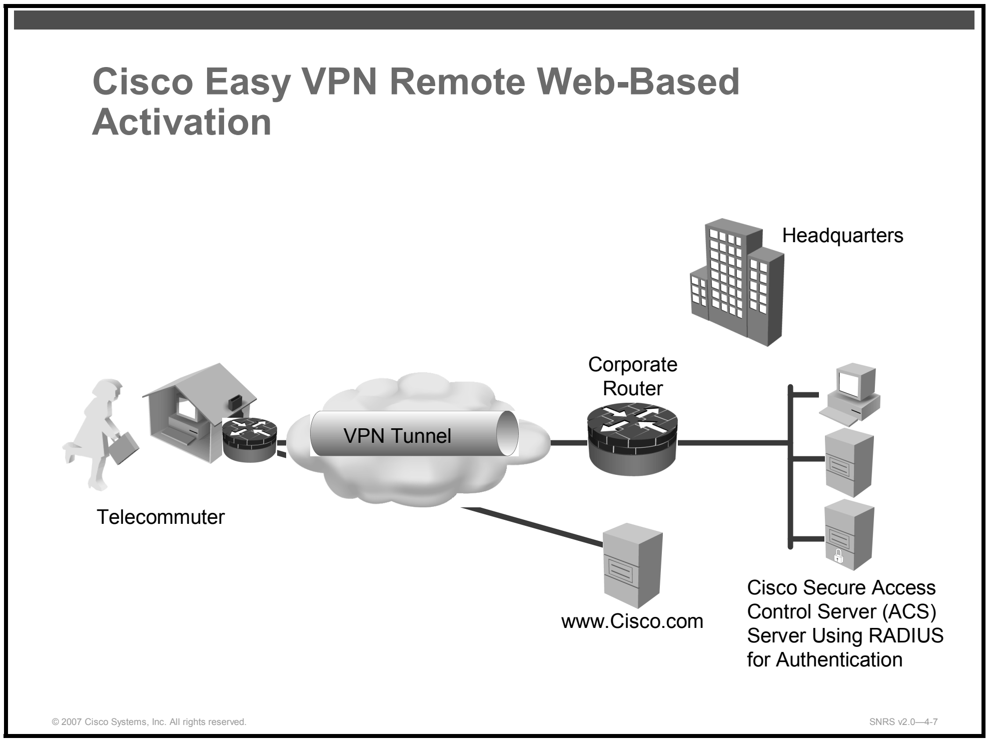
XAUTH is controlled by the Cisco Easy VPN Server. When the Cisco Easy VPN Server requests XAUTH authentication, the following messages are displayed on the console of the router:

```
EZVPN: Pending XAuth Request, Please enter the following command:
crypto ipsec client ezvpn xauth
```

When you see this message, you can provide the necessary user ID, password, and other information by entering the **crypto ipsec client ezvpn connect** command and responding to the prompts that follow.

The recommended XAUTH timeout is 50 seconds or less.

Note The timeout for entering the username and password is determined by the configuration of the Cisco Easy VPN Server. For servers running Cisco IOS Software, this timeout value is specified by the **crypto isakmp xauth timeout** command.



Cisco Easy VPN Remote Web-Based Activation

Headquarters
Corporate Router
VPN Tunnel
Telecommuter
www.Cisco.com
Cisco Secure Access Control Server (ACS) Server Using RADIUS for Authentication

SNRS v2.0—4-7

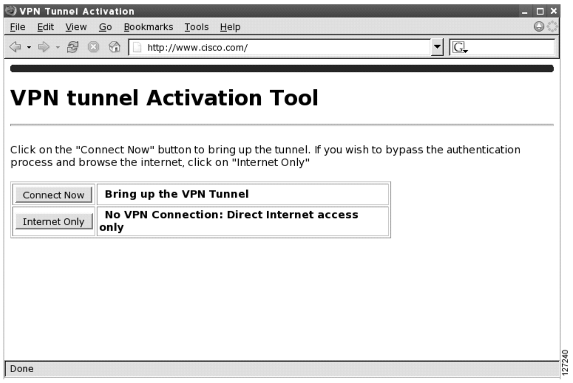
Cisco Easy VPN Remote Web-Based Activation

Cisco Easy VPN Remote Web-Based Activation provides a user-friendly method for remote teleworkers to authenticate the VPN tunnel between their Cisco Easy VPN Remote router and the central site router. This feature allows administrators to set up their remote LANs so that the initial HTTP request that is coming from any of the remote PCs is intercepted by the Cisco Easy VPN Remote router. A login page is returned to the user, whereby the user may enter credentials to authenticate the VPN tunnel. After the VPN tunnel comes up, all users behind this remote site can access the corporate LAN without being reprompted for the username and password. Alternatively, the user may choose to bypass the VPN tunnel and connect only to the Internet, in which case a password is not required.

A typical application for web-based activation is for home teleworkers who bring up the Cisco Easy VPN tunnel only when they need to connect to the corporate LAN. If the remote teleworker is not present, other members of the household (such as a spouse or children) can use the Internet Only option to browse the Internet without activating the VPN tunnel.

Note Entering the XAUTH credentials brings up the tunnel for all users who are behind this remote site. After the tunnel is up, any additional PCs that are behind the remote site do not get prompted for XAUTH credentials. Cisco Easy VPN Remote Web-Based Activation is an authentication to bring up the VPN tunnel for all remote PCs and cannot be considered individual user authentication. Individual user authentication for VPN tunnel access is available using the Cisco IOS authentication proxy or 802.1x features, which can be configured on the Cisco Easy VPN Remote router.

Web-Based Activation



VPN tunnel Activation Tool

Click on the "Connect Now" button to bring up the tunnel. If you wish to bypass the authentication process and browse the internet, click on "Internet Only"

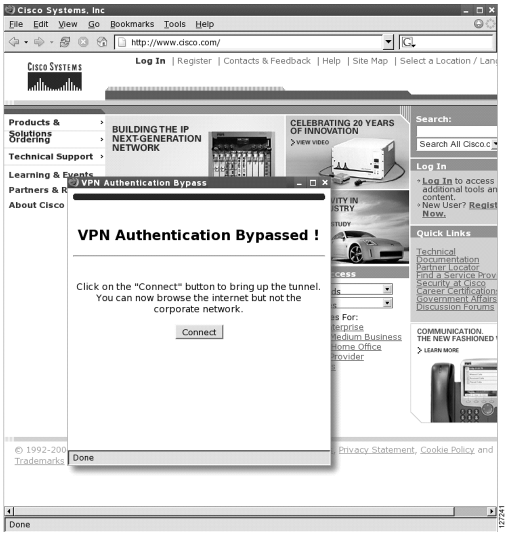
Connect Now	**Bring up the VPN Tunnel**
Internet Only	**No VPN Connection: Direct Internet access only**

SNRS v2.0—4-8

This figure is an example of a web-based activation portal page. Users may choose to connect to the corporate LAN by clicking Connect Now or they may choose to connect only to the Internet by clicking Internet Only.

Note	If the user chooses to connect only to the Internet, a password is not required.

Authentication Bypass



Cisco Systems, Inc
File Edit View Go Bookmarks Tools Help
http://www.cisco.com/
Log In | Register | Contacts & Feedback | Help | Site Map | Select a Location / Lang
Products & Solutions
Ordering
Technical Support
Learning & Events
About Cisco
BUILDING THE IP NEXT-GENERATION NETWORK
CELEBRATING 20 YEARS OF INNOVATION
Search:
Log In
Quick Links
VPN Authentication Bypass
Connect
Done

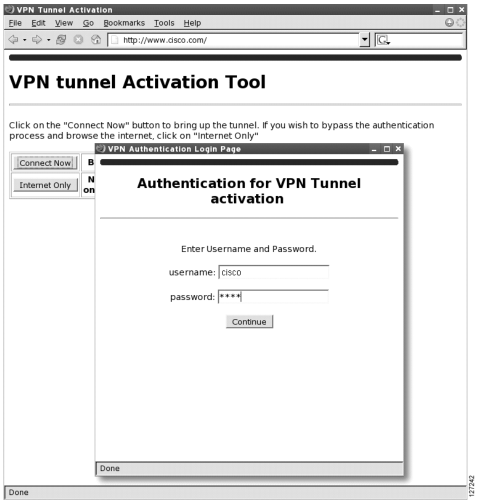
VPN Authentication Bypassed !

Click on the "Connect" button to bring up the tunnel. You can now browse the internet but not the corporate network.

This figure is an example of a web-based activation in which the user chose to connect only to the Internet by clicking the Internet Only option. This option is most useful for household members who need to browse the Internet while the remote teleworker is not available to authenticate the VPN tunnel for corporate use.

Note	If a user mistakenly closes the Web-Based Activation window, the window can be reopened by accessing the remote router (by entering http://routeripaddress/ezvpn/connect). After the Web-Based Activation window opens, the Cisco Easy VPN tunnel can be authenticated.

User Authentication



VPN tunnel Activation Tool

Click on the "Connect Now" button to bring up the tunnel. If you wish to bypass the authentication process and browse the internet, click on "Internet Only"

Authentication for VPN Tunnel activation

Enter Username and Password.

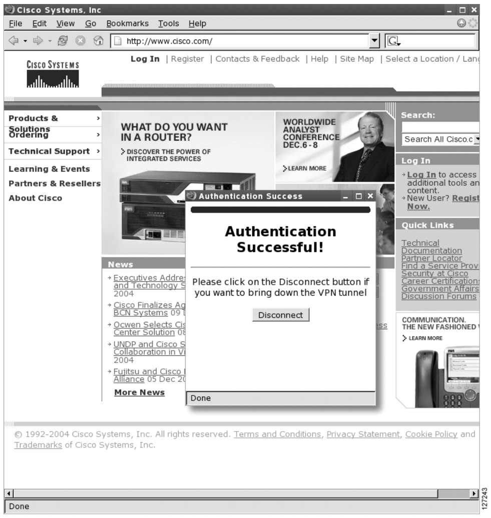
username: cisco

password: ****

SNRS v2.0—4-10

This figure is an example of a web-based activation in which the user chose to connect to the corporate LAN by entering a username and password. After the user is successfully authenticated, the Cisco Easy VPN tunnel is brought up for this remote site. If there are multiple PCs behind this remote site, none of the additional users who are connecting to the corporate LAN will be requested for the XAUTH credentials because the tunnel is already up.

Successful Authentication



Authentication Success

Authentication Successful!

Please click on the Disconnect button if you want to bring down the VPN tunnel

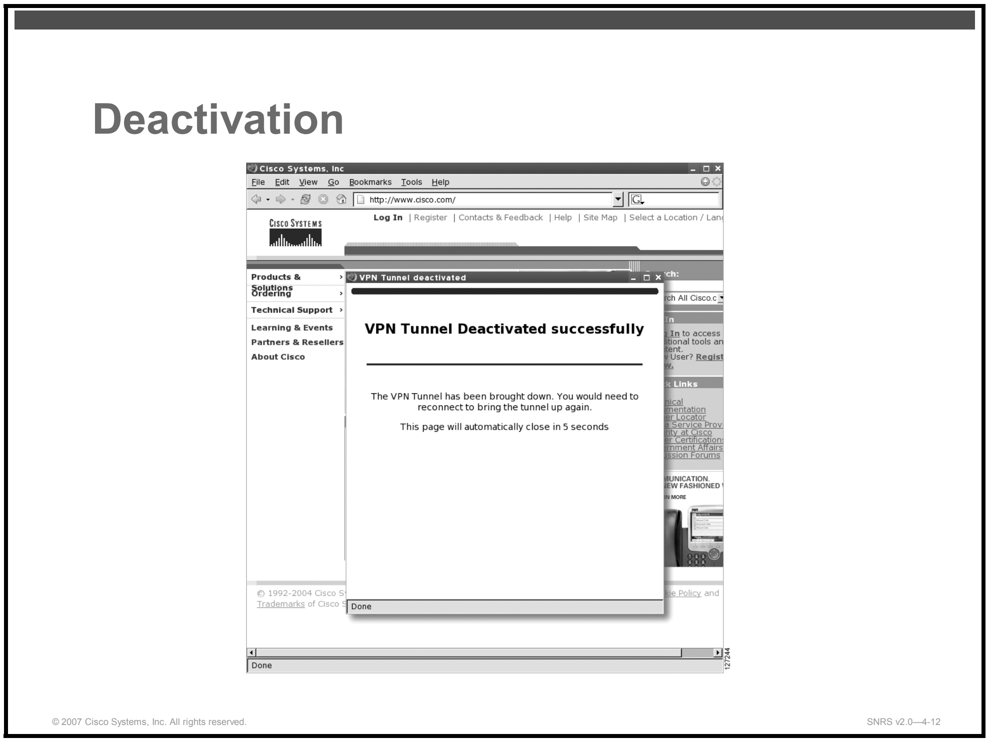
Disconnect

Done

Cisco Systems, Inc
File Edit View Go Bookmarks Tools Help
http://www.cisco.com/
Log In | Register | Contacts & Feedback | Help | Site Map | Select a Location / Lan
CISCO SYSTEMS
Products & Solutions
Ordering
Technical Support
Learning & Events
Partners & Resellers
About Cisco
WHAT DO YOU WANT IN A ROUTER?
DISCOVER THE POWER OF INTEGRATED SERVICES
WORLDWIDE ANALYST CONFERENCE DEC.6 - 8
LEARN MORE
Search:
Search All Cisco.c
Log In
Log In to access additional tools an content.
New User? Regist Now.
Quick Links
Technical Documentation
Partner Locator
Find a Service Prov
Security at Cisco
Career Certification
Government Affairs
Discussion Forums
News
Executives Addre and Technology S 2004
Cisco Finalizes Ag BCN Systems 09
Ocwen Selects Cis Center Solution 08
UNDP and Cisco S Collaboration in Vi 2004
Fujitsu and Cisco Alliance 05 Dec 20
More News
COMMUNICATION. THE NEW FASHIONED
LEARN MORE
© 1992-2004 Cisco Systems, Inc. All rights reserved. Terms and Conditions, Privacy Statement, Cookie Policy and Trademarks of Cisco Systems, Inc.
Done
127243

SNRS v2.0—4-11

This figure is an example of a successful activation. If the user chooses to deactivate the VPN tunnel, the user should click the Disconnect button. After the IKE SA times out (the default value is 24 hours), the remote teleworker has to enter the XAUTH credentials to bring up the tunnel.



Deactivation

VPN Tunnel Deactivated successfully

The VPN Tunnel has been brought down. You would need to reconnect to bring the tunnel up again.

This page will automatically close in 5 seconds

SNRS v2.0—4-12

This figure is an example of a VPN tunnel that has been deactivated successfully. The page automatically closes in 5 seconds.

802.1x Authentication

The 802.1x Authentication feature allows you to combine Cisco Easy VPN client mode operation with 802.1x authentication on Cisco IOS routers.

Tunnel Activation Options

There are three tunnel activation options:

- Automatic activation
- Manual activation
- Traffic-triggered activation (not available in Cisco IOS Release 12.3(11)T)

Tunnel connect and disconnect options are available with Cisco SDM.

Automatic Activation

The Cisco Easy VPN tunnel is automatically connected when the Cisco Easy VPN Remote feature is configured on an interface. If the tunnel times out or fails, the tunnel automatically reconnects and retries indefinitely.

To specify automatic tunnel control on a Cisco Easy VPN Remote device, you need to configure the **crypto ipsec client ezvpn** command and then the **connect auto** subcommand. However, you do not need to use these two commands when you are creating a new Cisco Easy VPN Remote configuration because the default is “automatic.”

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use Cisco SDM.

Manual Activation

The Cisco Easy VPN Remote software implements manual control of the Cisco Easy VPN tunnels so that you can establish and terminate the tunnel on demand.

To specify manual tunnel control on a Cisco Easy VPN Remote device, you need to input the **crypto ipsec client ezvpn** command and then the **connect manual** command.

The manual setting means that the Cisco Easy VPN Remote client router will wait for a command before attempting to establish the Cisco Easy VPN Remote connection. When the tunnel times out or fails, subsequent connections will also have to wait for the command.

If the configuration is manual, the tunnel is connected only after you issue the command **crypto ipsec client ezvpn connect**.

To disconnect or reset a particular tunnel, you should use the **clear crypto ipsec client ezvpn** command, or you can use Cisco SDM.

Traffic-Triggered Activation

The Traffic-Triggered Activation feature is recommended for transactional-based VPN applications. It is also recommended for use with the Cisco Easy VPN dial backup feature for the backup Cisco Easy VPN configuration so that backup is activated only when there is traffic to send across the tunnel.

To use ACL tunnel control, you must first describe the traffic that is considered "interesting."

To configure an ACL-triggered tunnel, use the **crypto ipsec client ezvpn** command with the **connect acl** subcommand.

Cisco Easy VPN Remote Features

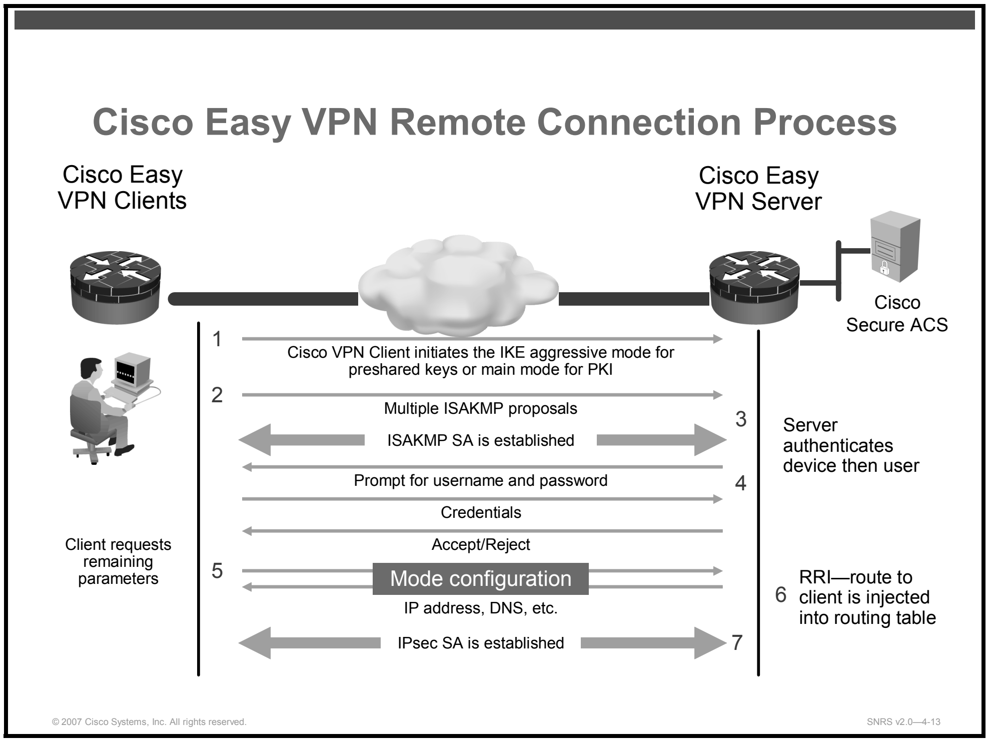
Cisco Easy VPN Remote is a collection of features that improves the capabilities of the Cisco Easy VPN Remote feature introduced in Cisco IOS Release 12.2(4)YA. Cisco Easy VPN Remote includes these features:

- **Default inside interface:** Cisco Easy VPN supports the automatic configuration of the default Cisco Easy VPN inside interface for Cisco 800 Series Routers.
- **Multiple inside interfaces:** It configures up to eight inside interfaces on the Cisco Easy VPN Remote.
- **Multiple outside interfaces:** It configures up to four outside tunnels for outside interfaces.
- **VLAN support:** It allows VLANs to be configured as valid Cisco Easy VPN inside interfaces.
- **Multiple subnet support:** It allows multiple subnets from the Cisco Easy VPN inside interface to be included in the Easy VPN tunnel.
- **NAT interoperability support:** It automatically restores the NAT configuration when the IPsec VPN tunnel is disconnected.
- **Local address support:** Cisco Easy VPN Remote is enhanced to support an additional **local-address** attribute that specifies which interface is used to determine the IP address used to source the Cisco Easy VPN tunnel traffic.
- **Peer hostname:** When a peer is defined as a hostname, the hostname is stored and the Domain Name System (DNS) lookup is done at the time of tunnel connection.
- **Proxy DNS server support:** It configures the router in a Cisco Easy VPN Remote configuration to act as a proxy DNS server for LAN-connected users.
- **Cisco IOS Firewall support:** It supports Cisco IOS Firewall configurations on all platforms.
- **Cisco Easy VPN Remote and Cisco Easy VPN Server on the same interface:** The Cisco Easy VPN Remote and Cisco Easy VPN Server are supported on the same interface, which makes it possible to establish a tunnel to another Cisco Easy VPN Server and terminate the Cisco Easy VPN Software Client on the same interface simultaneously.
- **Cisco Easy VPN Remote and site-to-site on the same interface:** Cisco Easy VPN Remote and site-to-site (crypto map) are supported on the same interface. This makes it possible to establish a tunnel to another Cisco Easy VPN Server and have another site-to-site on the same interface simultaneously.
- **Cisco Easy VPN Remote web managers:** Users can manage Cisco Easy VPN Remote on the Cisco uBR905 Cable Access Router and the Cisco uBR925 Cable Access Router using a built-in web interface.
- **IPsec Dead Peer Detection (DPD) Periodic Message Option:** This feature allows you to configure your router to query the status of its IKE peer at regular intervals.
- **Load balancing:** If a remote device is loaded and unable to accept more traffic, the Cisco VPN 3000 Concentrator will send a notify message that contains an IP address that represents the new IKE server to which the remote should connect.
- **Management enhancements:** It allows for remote management of the Cisco Easy VPN Remote device.
- **Perfect forward secrecy (PFS) support:** The PFS configuration mode attribute is sent by the server if requested by the Cisco Easy VPN Remote device.

- **Dial backup:** It allows you to configure a dial backup tunnel connection on your remote device.
- **Cisco Easy VPN Virtual Interface Support on a Server:** This feature allows you to selectively send traffic to different Cisco VPN concentrators and to the Internet (includes a reference to the IPsec Virtual Tunnel Interface feature.)
- **Cisco Easy VPN Remote Dual Tunnel Support:** This feature allows you to configure multiple Cisco Easy VPN tunnels that share common inside and outside interfaces to connect two peers to two different VPN servers simultaneously.
- **Banner:** The Cisco Easy VPN Remote device can download a banner that has been pushed by the Cisco Easy VPN Server. The banner can be used for XAUTH and web-based activation. The banner is displayed when the Cisco Easy VPN tunnel is up on the Cisco Easy VPN Remote console or as an HTML page in the case of web-based activation.
- **Configuration management enhancements (Pushing a Configuration URL Through a Mode-Configuration Exchange):** The Cisco Easy VPN Remote device can download a URL that is pushed by the Cisco Easy VPN Server, allowing the Cisco Easy VPN Remote device to download configuration content and apply it to the running configuration.
- **Reactivate Primary Peer:** This feature allows you to designate a primary peer. When a Cisco Easy VPN remote device fails over from the primary peer to a backup peer and the primary peer is again available, connections with the backup peer are torn down and a connection is made with the primary peer.

How Cisco Easy VPN Works

This topic describes the operations of Cisco Easy VPN.



Cisco Easy VPN Remote Connection Process

Cisco Easy VPN Clients

Cisco Easy VPN Server

Cisco Secure ACS

1 Cisco VPN Client initiates the IKE aggressive mode for preshared keys or main mode for PKI

2 Multiple ISAKMP proposals

ISAKMP SA is established

3 Server authenticates device then user

Prompt for username and password

4 Credentials

Accept/Reject

Client requests remaining parameters

5 Mode configuration

IP address, DNS, etc.

6 RRI—route to client is injected into routing table

IPsec SA is established 7

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-13

When a Cisco Easy VPN Remote client initiates a connection with a Cisco Easy VPN Server gateway, the “conversation” that occurs between the peers generally consists of the following major steps:

- Device authentication via ISAKMP
- User authentication using IKE XAUTH
- VPN policy push (using mode configuration)
- IPsec SA creation

The following is a detailed description of the Cisco Easy VPN Remote connection process:

1. The Cisco VPN Client initiates the IKE Phase 1 process.
2. The Cisco VPN Client establishes an ISAKMP SA.
3. The Cisco Easy VPN Server accepts the SA proposal.
4. The Cisco Easy VPN Server initiates a username and password challenge.
5. The mode configuration process is initiated.
6. The Reverse Route Injection (RRI) process is initiated.
7. IPsec quick mode completes the connection.

Authentication Begins

Because there are two ways to perform authentication, the Cisco VPN Client must consider the following when initiating this phase:

- If a pre-shared key is to be used for authentication, the Cisco VPN Client initiates aggressive mode. When pre-shared keys are used, the accompanying group name entered in the configuration GUI (ID_KEY_ID) is used to identify the group profile associated with this Cisco VPN Client.
- If digital certificates are to be used for authentication, the Cisco VPN Client initiates main mode. When digital certificates are used, the Organizational Unit (OU) field of a distinguished name (DN) is used to identify the group profile.

Because the Cisco VPN Client may be configured for pre-shared key authentication, which initiates IKE aggressive mode, it is recommended that the administrator change the identity of the Cisco Easy VPN remote device via the **crypto isakmp identity hostname** command. This action does not affect certificate authentication via IKE main mode.

An ISAKMP SA Is Established

- The Cisco VPN Client attempts to establish an SA between peer IP addresses by sending multiple ISAKMP proposals to the Cisco Easy VPN Server.
- To reduce manual configuration on the Cisco VPN Client, the ISAKMP proposals include several combinations of the following:
  - Encryption and hash algorithms
  - Authentication methods
  - DH group sizes

Cisco Easy VPN Server Authenticates the Device

The Cisco Easy VPN Server authenticates the device first before authenticating the user.

ISAKMP policy is global for the Cisco Easy VPN Server and can consist of several proposals. In the case of multiple proposals, the Cisco Easy VPN Server will use the first match (so you should always have your most secure policies listed first).

- The Cisco Easy VPN Server searches for a match.
  - The first proposal to match the server list is accepted (highest-priority match).
  - The most secure proposals are always listed at the top of the Cisco Easy VPN Server proposal list (highest priority).
- The ISAKMP SA is successfully established.
- Device authentication ends and user authentication begins.

Username and Password Challenge Is Processed

If the Cisco Easy VPN Server is configured for XAUTH, the Cisco VPN Client waits for a username and password challenge:

- The user enters a username and password combination.
- The username and password information is checked against authentication entities using AAA.

The information that is entered is checked against authentication entities using AAA protocols such as RADIUS and TACACS+. Token cards may also be used via AAA proxy.

Cisco Easy VPN devices that are configured to handle remote VPN Clients should always be configured to enforce user authentication.

Mode Configuration

If the Cisco Easy VPN Server indicates successful authentication, the Cisco VPN Client requests the remaining configuration parameters from the Cisco Easy VPN Server using mode configuration.

The remaining system parameters (IP address, DNS, split tunnel attributes, and so on) are pushed to the Cisco VPN Client at this time using mode configuration.

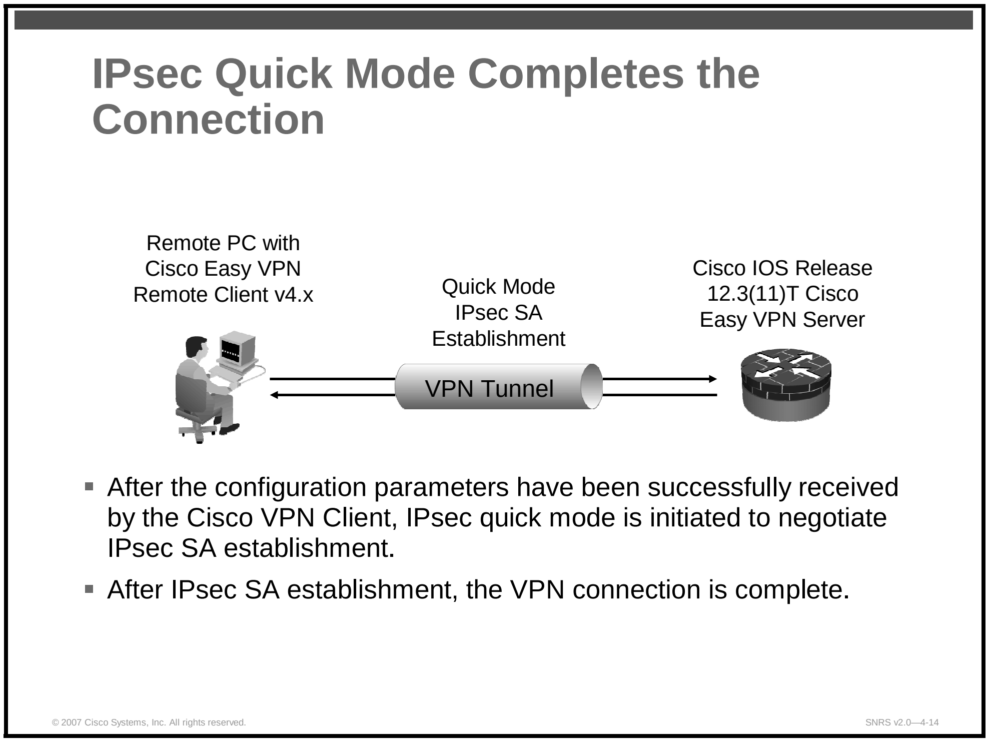
Note The IP address is the only required parameter in a group profile; all other parameters are optional

RRI Process Is Initiated

After the Cisco Easy VPN Server knows the assigned IP address of the Cisco VPN Client, it must determine how to route packets through the appropriate VPN tunnel.

RRI ensures that a static route is created on the Cisco Easy VPN Server for the internal IP address of each Cisco VPN Client.

Note It is recommended that you enable RRI on the crypto map (static or dynamic) for the support of Cisco VPN Clients, unless the crypto map is being applied to a generic routing encapsulation (GRE) tunnel that is already being used to distribute routing information.



IPsec Quick Mode Completes the Connection

- After the configuration parameters have been successfully received by the Cisco VPN Client, IPsec quick mode is initiated to negotiate IPsec SA establishment.
- After IPsec SA establishment, the VPN connection is complete.

Connection Is Completed with IPsec Quick Mode

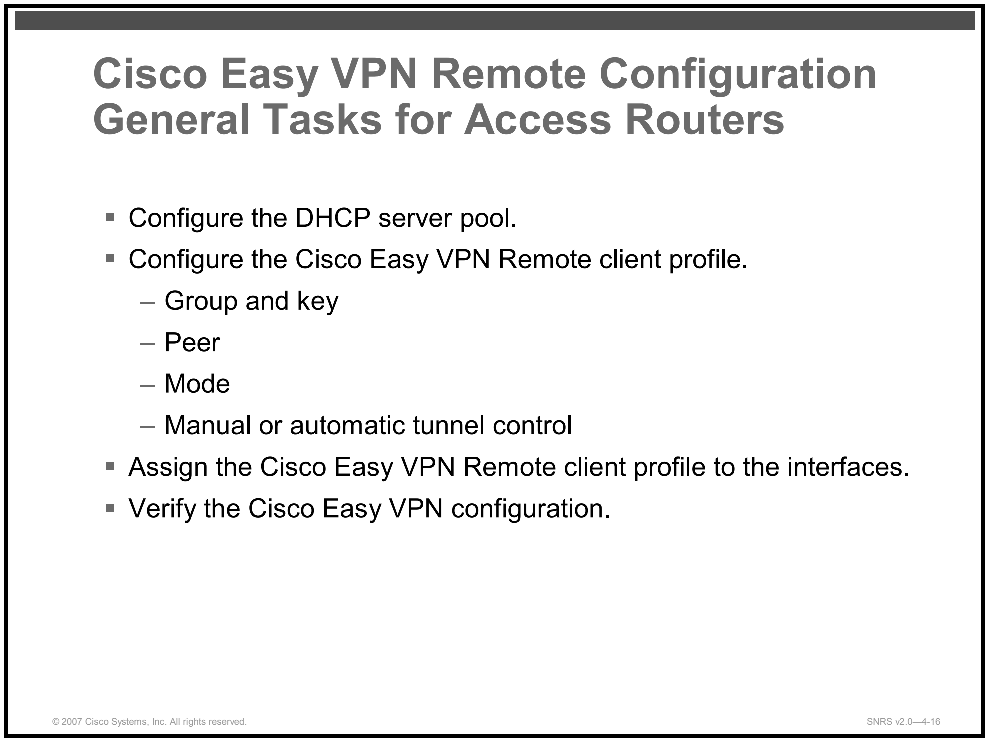
IKE Phase 2 has one mode, called quick mode. Quick mode occurs after IKE has established the secure tunnel in IKE Phase 1. IKE Phase 2 negotiates a shared IPsec policy, derives shared-secret keying material used for the IPsec security algorithms, and establishes IPsec SAs. Quick mode exchanges nonces that provide replay protection. The nonces are used to generate new shared-secret key material and prevent replay attacks from generating bogus SAs.

Quick mode is also used to renegotiate a new IPsec SA when the IPsec SA lifetime expires. Base quick mode is used to refresh the keying material used to create the shared-secret key based on the keying material derived from the DH exchange in IKE Phase 1.

After IPsec SAs have been created, the connection is complete.

Configuring Cisco Easy VPN Remote for Access Routers

This topic describes how to configure a router as a Cisco Easy VPN Remote access client.



Cisco Easy VPN Remote Configuration General Tasks for Access Routers

- Configure the DHCP server pool.
- Configure the Cisco Easy VPN Remote client profile.
  - Group and key
  - Peer
  - Mode
  - Manual or automatic tunnel control
- Assign the Cisco Easy VPN Remote client profile to the interfaces.
- Verify the Cisco Easy VPN configuration.

SNRS v2.0—4-16

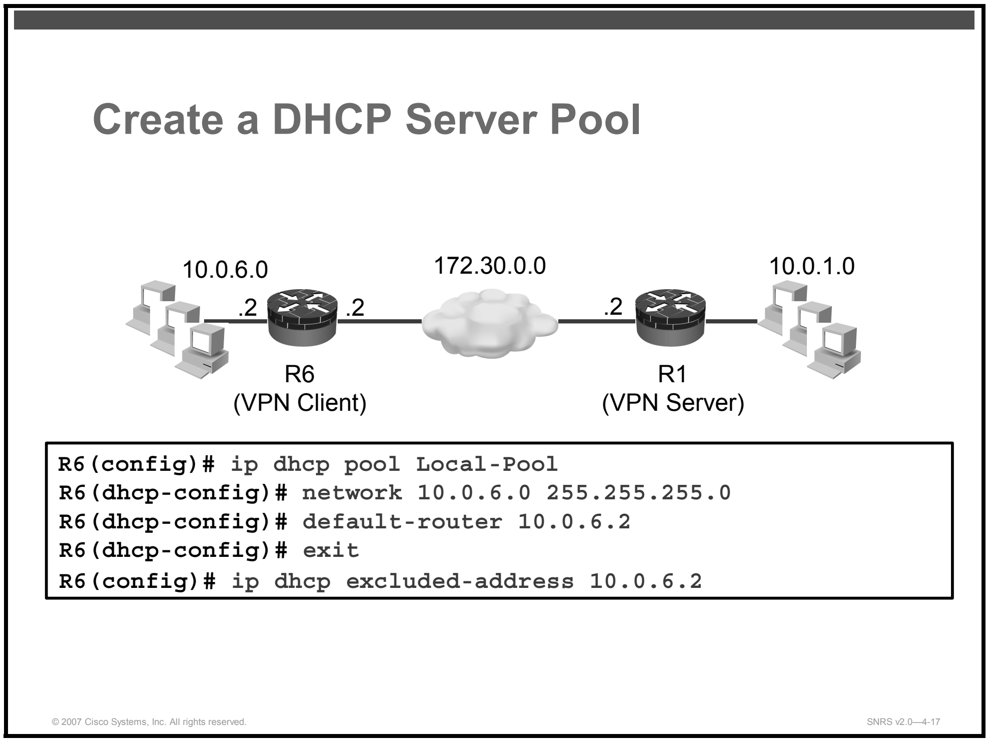
An access router at a remote site can be configured as a Cisco Easy VPM remote client. As a remote client, the access router can give out DHCP addresses to hosts behind it or you can let the Easy VPN server give out IP addresses to local hosts. The remote client profile specifies group names as well as VPN keys, peer IP addresses, mode of operation (i.e. client or network extension), and how to connect or initiate the tunnel (i.e. manual or auto).

Configuring Cisco access routers to act as Cisco Easy VPN Remote clients consists of the following tasks:

- (Optional) Configure the DHCP server pool.
- Configure the Cisco Easy VPN Remote client profile.
- Assign the Cisco Easy VPN Remote client profile to the interfaces.
- (Optional) Configure XAUTH Save Password.
- Initiate the VPN tunnel.
- Verify the Cisco Easy VPN configuration.

Configure the DHCP Server Pool

This topic describes how to configure a local DHCP pool for use in Cisco Easy VPN.



Create a DHCP Server Pool

```
R6(config)# ip dhcp pool Local-Pool
R6(dhcp-config)# network 10.0.6.0 255.255.255.0
R6(dhcp-config)# default-router 10.0.6.2
R6(dhcp-config)# exit
R6(config)# ip dhcp excluded-address 10.0.6.2
```

If you want to use the local router DHCP server to assign IP addresses to the hosts that are connected to the LAN interface of the router, you must create a pool of IP addresses for the onboard DHCP server of the server. The DHCP server then assigns an IP address from this pool to each host when it connects to the router.

In a typical VPN connection, the hosts connected to the router LAN interface are assigned an IP address in a private address space. The router then uses NAT or PAT to translate those IP addresses into a single IP address that is transmitted across the VPN tunnel connection.

Here are the steps to create the DHCP server pool:

Step 1 Create a DHCP server address pool and enter DHCP pool configuration mode.

```
R6(config)# ip dhcp pool <name>
```

Step 2 Specify the IP network and subnet mask of the address pool that will be used by the hosts connected to the local Ethernet interface of the router.

```
R6(dhcp-config)# network network-number [mask | /prefix-
length]
```

Step 3 Specify the IP address of the default router for a DHCP client. You must specify at least one address. You can optionally specify up to eight addresses per command.

```
R6(dhcp-config)# default-router address [address2 ...
address8]
```

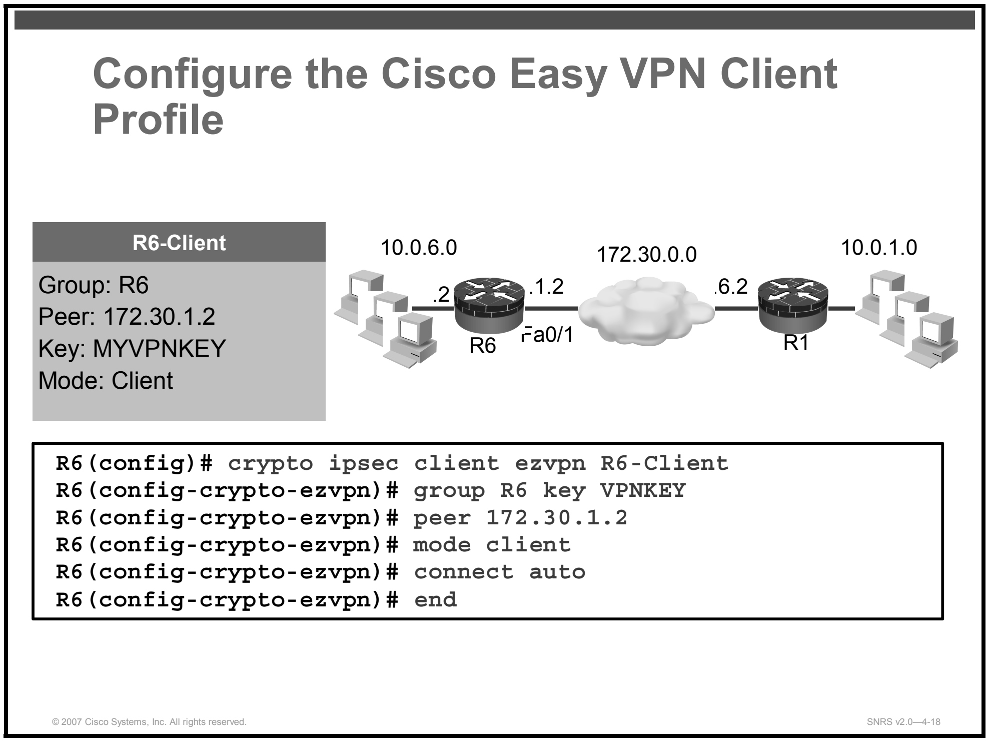
After a DHCP client has booted, the client begins sending packets to its default router. The IP address should be on the same subnet as the client. One IP address is required; however, you can specify up to eight addresses in one command line.

Step 4 Use the **ip dhcp excluded-address** command to exclude any addresses from the DHCP server pool. The *address* value should be the IP address assigned to the router LAN interface.

```
R6(config)# ip dhcp excluded-address low-address [high-address]
```

Configure and Assign the Cisco VPN Client Profile

This topic describes how to configure and assign the Cisco Easy VPN client profile.



Configure the Cisco Easy VPN Client Profile

R6-Client
Group: R6 Peer: 172.30.1.2 Key: MYVPNKEY Mode: Client

```
R6(config)# crypto ipsec client ezvpn R6-Client
R6(config-crypto-ezvpn)# group R6 key VPNKEY
R6(config-crypto-ezvpn)# peer 172.30.1.2
R6(config-crypto-ezvpn)# mode client
R6(config-crypto-ezvpn)# connect auto
R6(config-crypto-ezvpn)# end
```

SNRS v2.0—4-18

To configure the Cisco Easy VPN client profile and to assign the profile to a router interface follow these steps:

Step 1 Create a remote configuration and enter Cisco Easy VPN Remote configuration mode.

```
R6(config)# crypto ipsec client ezvpn name
```

Step 2 Specify the IPsec group and IPsec key values to be associated with this profile.

```
R6(config-crypto-ezvpn)# group group-name key group-key
```

Note The values of *group-name* and *group-key* must match the values assigned in the Cisco Easy VPN Server. The value of the *group-key* argument must match the key defined on the Cisco Easy VPN Server.

Step 3 Specify the IP address or hostname for the destination peer. This is typically the IP address of the Cisco Easy VPN Server router outside interface. If you prefer to specify a hostname, you must have a DNS server configured and available.

```
R6(config-crypto-ezvpn)# peer [ip-address | hostname]
```

Step 4 Specify the type of VPN connection that should be made (client or network extension).

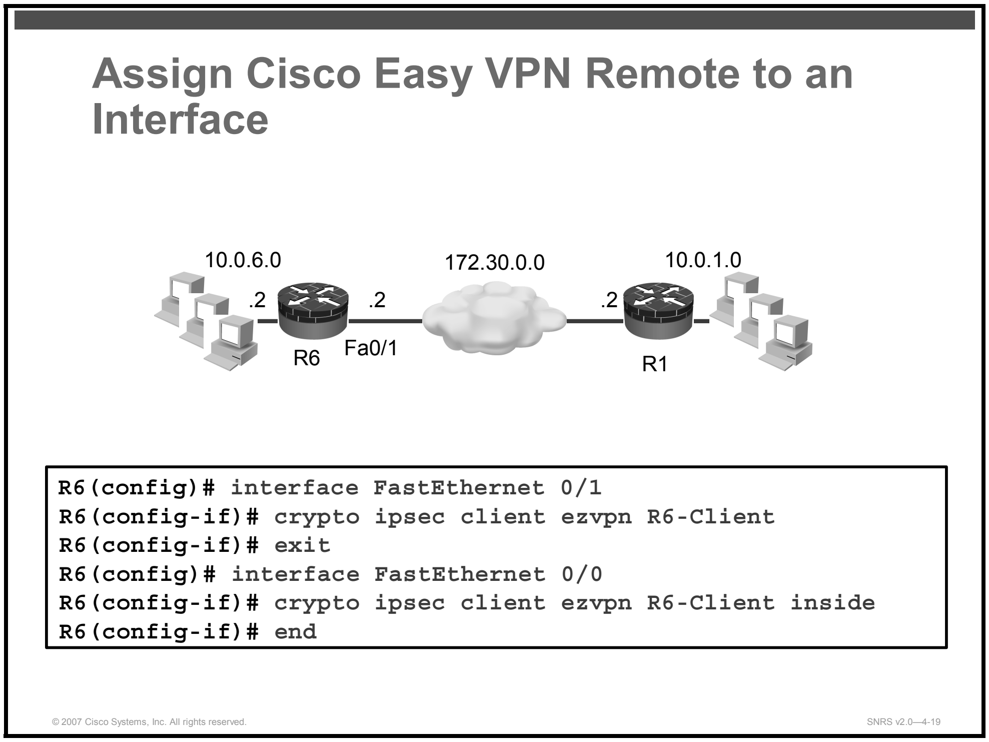
```
R6(config-crypto-ezvpn)# mode {client | network-extension |
network-plus}
```

Step 5 Specify manual or automatic connections.

```
R6(config-crypto-ezvpn)# connect [ acl | auto | manual ]
```

Note Automatic is the default; you do not need to use the **manual** keyword if your configuration is automatic.

To connect to a specified IPsec VPN tunnel in a manual configuration, use the **manual** keyword.



Assign Cisco Easy VPN Remote to an Interface

```
R6(config)# interface FastEthernet 0/1
R6(config-if)# crypto ipsec client ezvpn R6-Client
R6(config-if)# exit
R6(config)# interface FastEthernet 0/0
R6(config-if)# crypto ipsec client ezvpn R6-Client inside
R6(config-if)# end
```

SNRS v2.0—4-19

All that remains is to assign the Cisco Easy VPN Remote client profile to an interface.

Use these steps to assign a client profile to an interface.

Step 1 Change to interface configuration mode.

```
R6(config)# interface interface
```

Step 2 Assign a Cisco Easy VPN Remote client profile to a router interface.

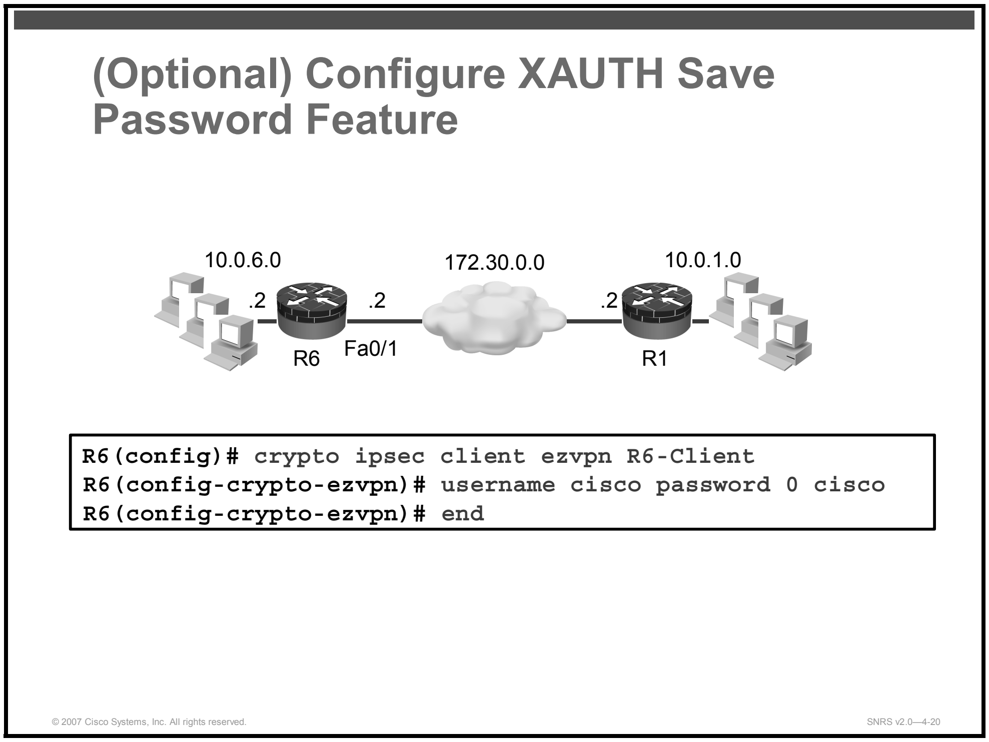
```
R6(config-if)# crypto ipsec client ezvpn name [outside |
inside]
```

Syntax Description

name	This specifies the Cisco Easy VPN Remote configuration to be assigned to the interface.
outside	(Optional) This specifies the outside interface of the IPsec client router. You can add up to four outside tunnels for all platforms, one tunnel per outside interfaces.
inside	(Optional) This specifies the inside interface of the IPsec client router. The Cisco 1700 Series Modular Access Routers have no default inside interface and any inside interface must be configured. The Cisco 800 Series Routers and the Cisco uBR905 and Cisco uBR925 Cable Access Routers have default inside interfaces. However, you can configure any inside interface. You can add up to three inside interfaces for all platforms.

Configure XAUTH Password Save

This topic describes how to configure XAUTH.



(Optional) Configure XAUTH Save Password Feature

```
R6(config)# crypto ipsec client ezvpn R6-Client
R6(config-crypto-ezvpn)# username cisco password 0 cisco
R6(config-crypto-ezvpn)# end
```

SNRS v2.0—4-20

This is an optional task. If you are not using XAUTH, skip this task.

Note If you have the save password feature enabled in the Cisco Easy VPN Server, you must enable it on the client as well. If both ends of the tunnel do not match, the VPN tunnel will not be established.

This task could be done as part of configuring the Cisco Easy VPN Remote client profile, to speed up the entry process.

Complete these steps to enable the Save Password feature.

Step 1 Enter the Cisco Easy VPN Remote configuration mode.

```
R6(config)# crypto ipsec client ezvpn name
```

Step 2 Save your XAUTH password locally.

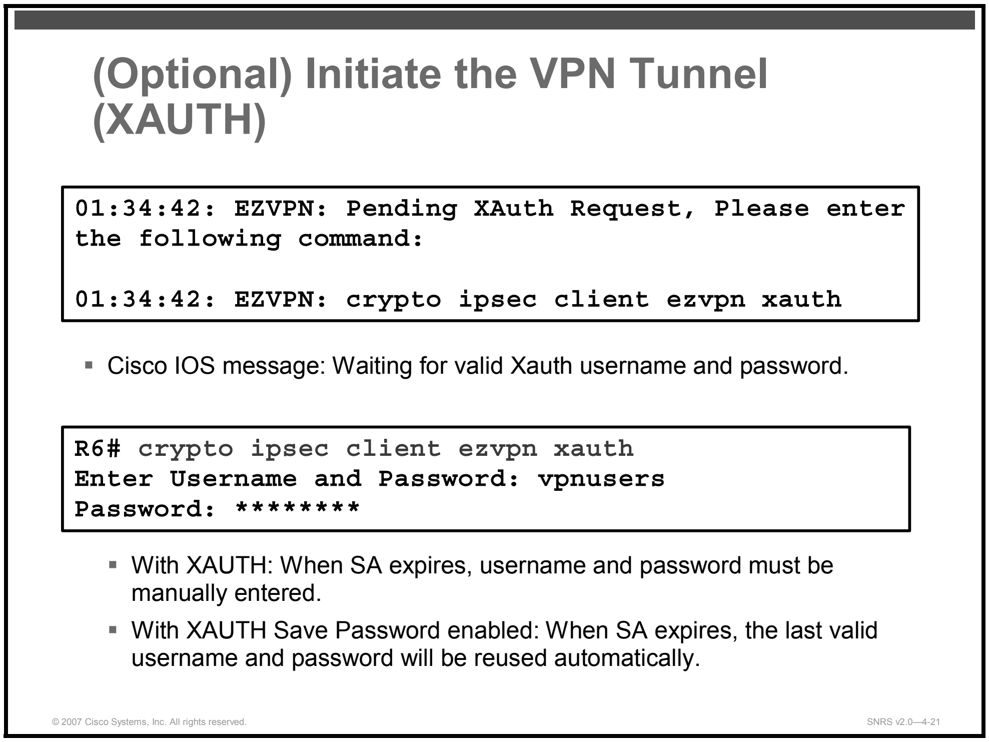
```
R6(config-crypto-ezvpn)# username cisco password 0 cisco
```

Enter the **username** command in Cisco Easy VPN Remote configuration mode for the specific client profile, as shown in the figure. This is the AAA username and password used to automatically reauthenticate the user with the XAUTH Save Password feature enabled in Cisco Easy VPN Server.

If XAUTH is used, no additional configuration is required on the client end. If the router is not configured with a saved username and password, the client router will prompt users to enter their username and password. To enter XAUTH credentials, users will need to enter **crypto ipsec client ezvpn xauth** at the command line and enter their username and password when prompted.

Initiating the VPN Tunnel

This topic describes how to manually initiate the VPN tunnel.



(Optional) Initiate the VPN Tunnel (XAUTH)

```
01:34:42: EZVPN: Pending XAuth Request, Please enter
the following command:

01:34:42: EZVPN: crypto ipsec client ezvpn xauth
```

- Cisco IOS message: Waiting for valid Xauth username and password.

```
R6# crypto ipsec client ezvpn xauth
Enter Username and Password: vpnusers
Password: ********
```

- With XAUTH: When SA expires, username and password must be manually entered.
- With XAUTH Save Password enabled: When SA expires, the last valid username and password will be reused automatically.

SNRS v2.0—4-21

This task is also optional. If you are not using XAUTH, skip this task.

With XAUTH configured, you must initiate the VPN tunnel manually (at least for the first time). The Cisco IOS Software message shown in the figure is displayed because the software is waiting for a valid XAUTH username and password. You will see this message whenever you log in to the remote router console port.

To connect to a specified IPsec VPN tunnel in a manual configuration, use the **crypto ipsec client ezvpn connect** command. To initial the VPN tunnel, complete these steps:

Step 1 Enter the **crypto ipsec client ezvpn xauth** command.

```
R6# crypto ipsec client ezvpn [connect | xauth]
```

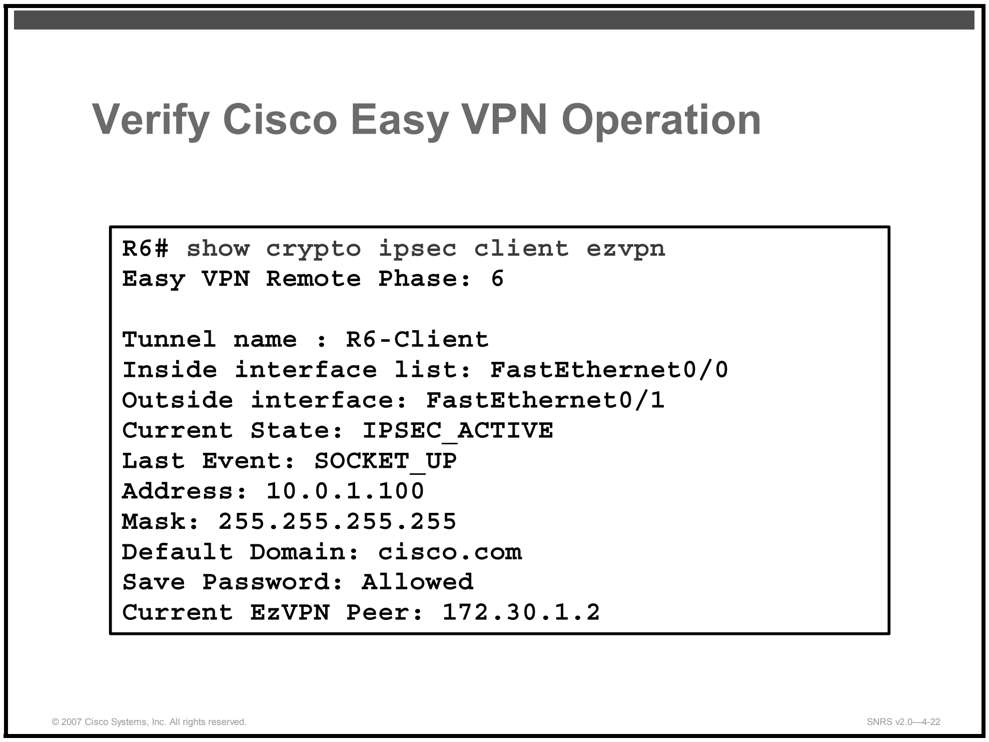
Step 2 For XAUTH enter the username and password as prompted.

Which of these two options happens next is determined by the XAUTH configuration:

- With just the XAUTH feature enabled, when the SA expires, you must manually re-enter the username and password. This process is ongoing. You will see the same Cisco IOS message and will have to repeat this manual process to reauthenticate each time.
- With XAUTH Save Password enabled, when the SA expires, the last valid username and password will be reused automatically. This option is the more popular of the two.

Verify the Cisco Easy VPN Configuration

This topic describes how to verify the Cisco Easy VPN configuration on the router.

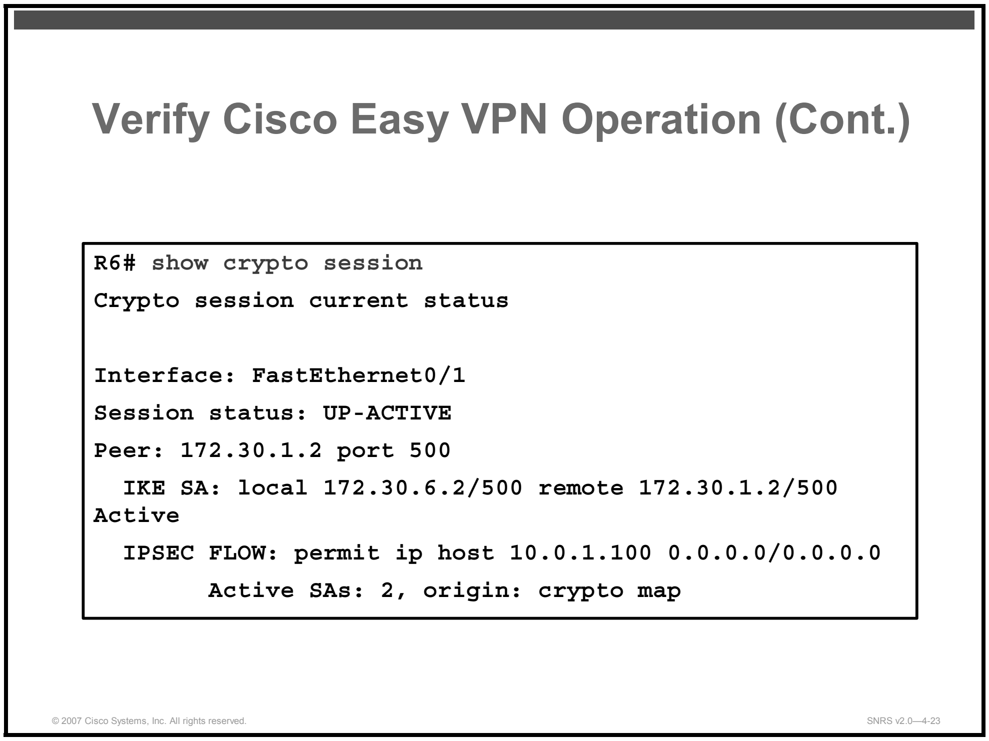


Verify Cisco Easy VPN Operation

```
R6# show crypto ipsec client ezvpn
Easy VPN Remote Phase: 6

Tunnel name : R6-Client
Inside interface list: FastEthernet0/0
Outside interface: FastEthernet0/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.1.100
Mask: 255.255.255.255
Default Domain: cisco.com
Save Password: Allowed
Current EzVPN Peer: 172.30.1.2
```

SNRS v2.0—4-22

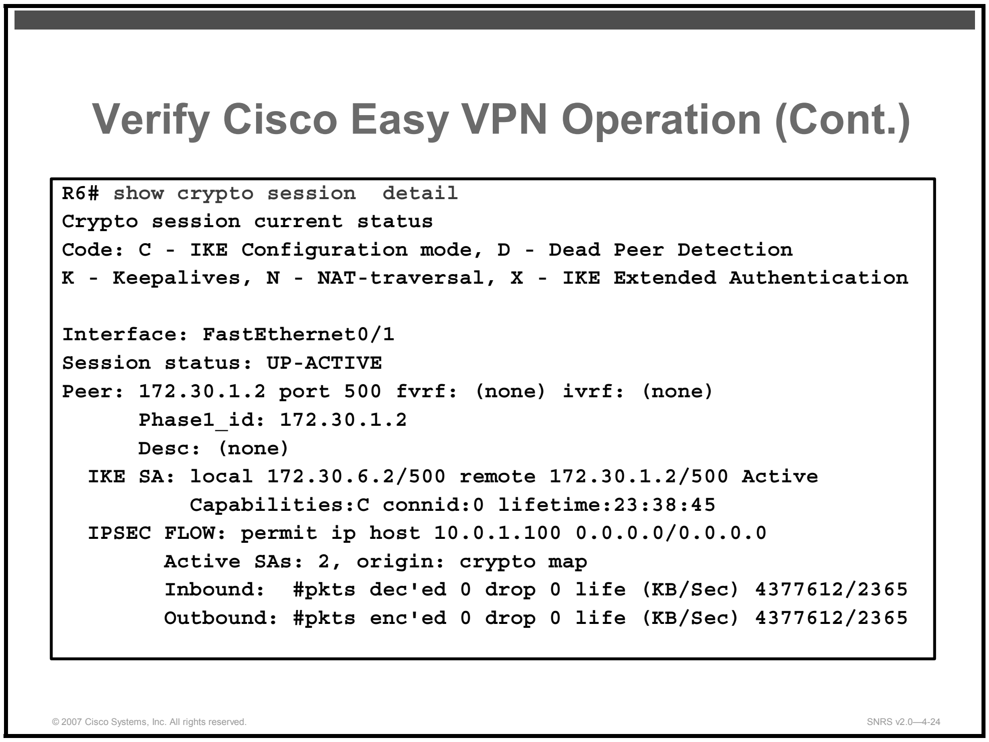
Verifying Cisco Easy VPN configuration consists of using various **show** commands. Examples are the **show crypto ipsec client ezvpn** and **show crypto session** commands**.**



Verify Cisco Easy VPN Operation (Cont.)

```
R6# show crypto session
Crypto session current status


Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.30.1.2 port 500
  IKE SA: local 172.30.6.2/500 remote 172.30.1.2/500
Active
  IPSEC FLOW: permit ip host 10.0.1.100 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
```

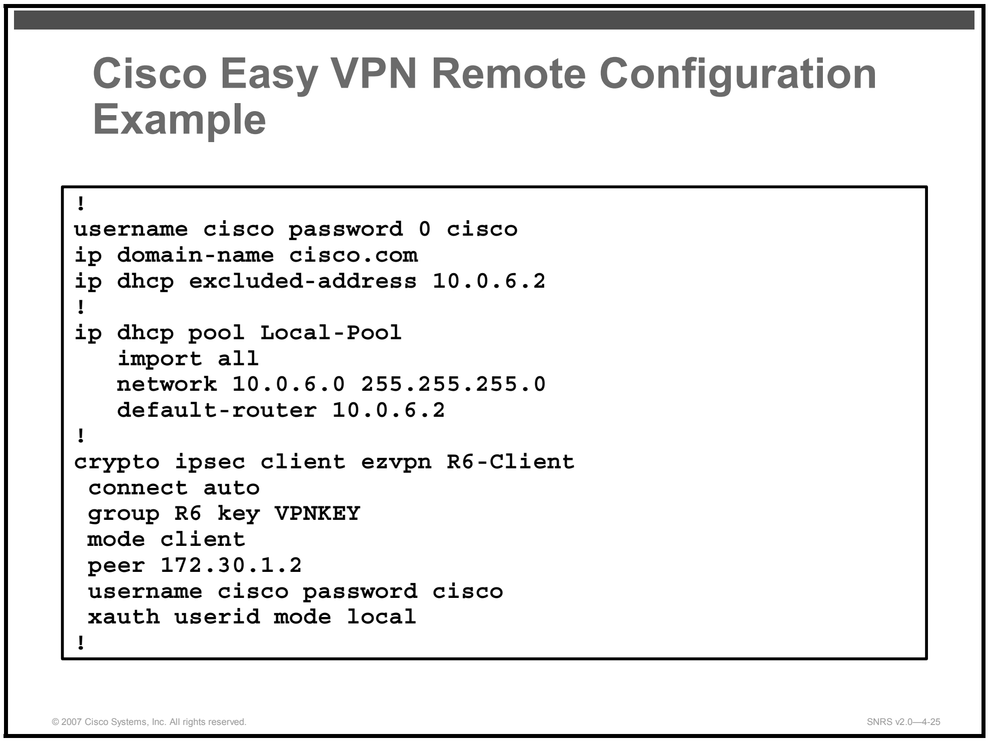
SNRS v2.0—4-23



Verify Cisco Easy VPN Operation (Cont.)

```
R6# show crypto session  detail
Crypto session current status
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, X - IKE Extended Authentication

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.30.1.2 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 172.30.1.2
      Desc: (none)
  IKE SA: local 172.30.6.2/500 remote 172.30.1.2/500 Active
          Capabilities:C connid:0 lifetime:23:38:45
  IPSEC FLOW: permit ip host 10.0.1.100 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 4377612/2365
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 4377612/2365
```

This figure shows the output of the **show crypto session detail** command.

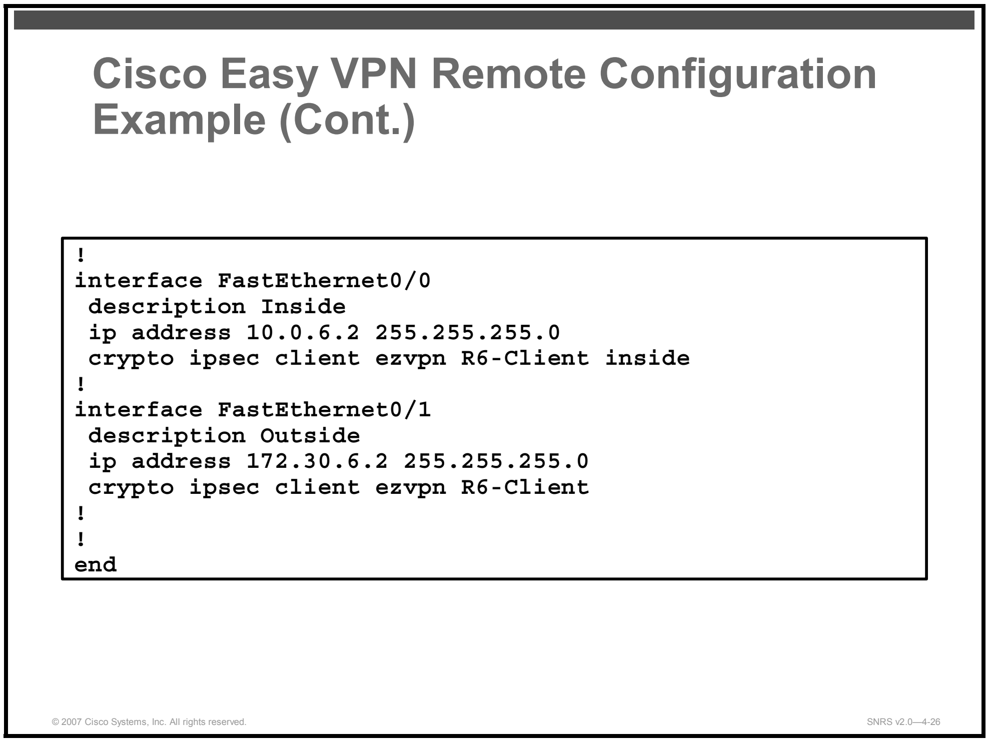


Cisco Easy VPN Remote Configuration Example

```
!
username cisco password 0 cisco
ip domain-name cisco.com
ip dhcp excluded-address 10.0.6.2
!
ip dhcp pool Local-Pool
   import all
   network 10.0.6.0 255.255.255.0
   default-router 10.0.6.2
!
crypto ipsec client ezvpn R6-Client
 connect auto
 group R6 key VPNKEY
 mode client
 peer 172.30.1.2
 username cisco password cisco
 xauth userid mode local
!
```

SNRS v2.0—4-25

This figure and the next detail an example of Cisco Easy VPN Remote access router configuration.



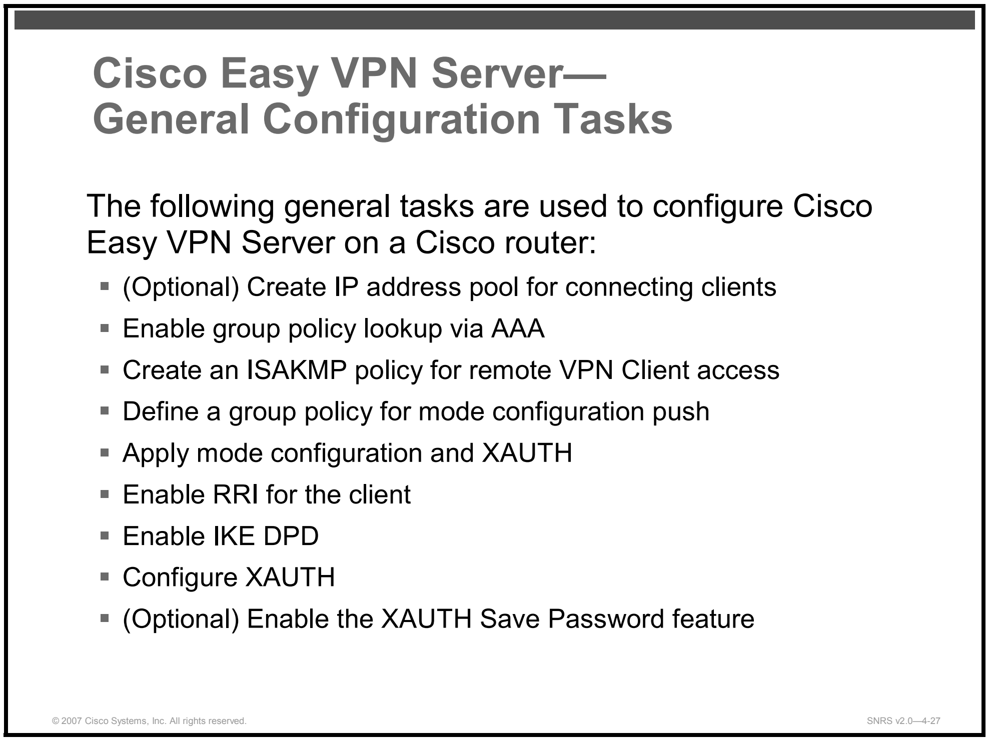
Cisco Easy VPN Remote Configuration Example (Cont.)

```
!
interface FastEthernet0/0
 description Inside
 ip address 10.0.6.2 255.255.255.0
 crypto ipsec client ezvpn R6-Client inside
!
interface FastEthernet0/1
 description Outside
 ip address 172.30.6.2 255.255.255.0
 crypto ipsec client ezvpn R6-Client
!
!
end
```

SNRS v2.0—4-26

Configuring Cisco Easy VPN Server

This topic describes how to configure a Cisco Easy VPN Server to support Cisco Easy VPN Remote client access.



Cisco Easy VPN Server—General Configuration Tasks

The following general tasks are used to configure Cisco Easy VPN Server on a Cisco router:

- (Optional) Create IP address pool for connecting clients
- Enable group policy lookup via AAA
- Create an ISAKMP policy for remote VPN Client access
- Define a group policy for mode configuration push
- Apply mode configuration and XAUTH
- Enable RRI for the client
- Enable IKE DPD
- Configure XAUTH
- (Optional) Enable the XAUTH Save Password feature

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-27

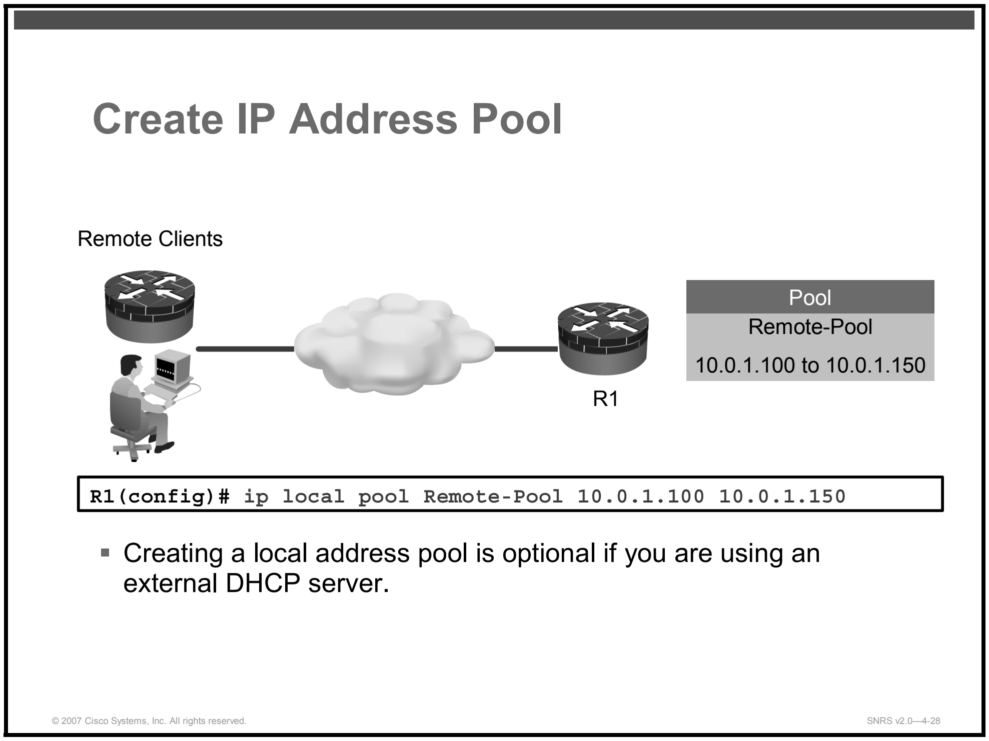
The Cisco Easy VPN Server feature allows a remote end user to communicate using IP Security (IPsec) with any Cisco IOS Virtual Private Network (VPN) gateway. Centrally managed IPsec policies are "pushed" to the client device by the server, minimizing configuration by the end user.

Complete the following tasks to configure a Cisco Easy VPN Server for XAUTH with Cisco Easy VPN Remote clients:

- Create IP address pool
- Configure group policy lookup
- Create an ISAKMP policy for remote VPN client access
- Define a group policy for a mode configuration push
- Create a transform set
- Create a dynamic crypto map with RRI
- Apply a mode configuration to the dynamic crypto map
- Apply the crypto map to the router interface
- Enable ISAKMP DPD
- (Optional) Configure XAUTH. (XAUTH is not required when using Cisco Easy VPN but it is covered here as part of this example. This option can be disabled.)
- (Optional) Enable the XAUTH Save Password feature

Create IP Address Pool

This section describes how to create an IP address pool to use as addresses for connecting Cisco Easy VPN Remote clients.



Create IP Address Pool

```
R1(config)# ip local pool Remote-Pool 10.0.1.100 10.0.1.150
```

- Creating a local address pool is optional if you are using an external DHCP server.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-28

If you are using a local IP address pool, you will need to configure that pool using the **ip local pool** command.

To configure a local pool of IP addresses to be used when a remote peer connects to a point-to-point interface, complete these steps.

Step 1 Create a local pool and define the range of addresses for clients.

```
R1(config)# ip local pool {default | poolname} [low-ip-address
[high-ip-address]] [group group-name] [cache-size size]
```

Syntax Description

default	Creates a default local IP address pool that is used if no other pool is named
poolname	Name of the local IP address pool
low-IP-address **[***high-IP-address***]**	First and, optionally, last address in an IP address range
group *group-name*	(Optional) Creates a pool group
cache-size *size*	(Optional) Sets the number of IP address entries on the free list that the system checks before assigning a new IP address Returned IP addresses are placed at the end of the free list. Before assigning a new IP address to a user, the system checks the number of entries from the end of the list (as defined by the **cache-size** *size* option) to determine that there are no returned IP addresses for that user. The range for the cache size is 0 to 100. The default cache size is 20.

Use the **ip local pool** command to create one or more local address pools from which IP addresses are assigned when a peer connects. You may also add another range of IP addresses to an existing pool. To use a named IP address pool on an interface, use the **peer default ip address pool** interface configuration command. A pool name can also be assigned to a specific user by using AAA RADIUS and TACACS functions.

If no named local IP address pool is created, a default address pool is used on all point-to-point interfaces after the **ip address-pool local** global configuration command is issued. If no explicit IP address pool is assigned, but pool use is requested by use of the **ip address-pool local** command, the special pool named "default" is used.

The optional **group** keyword and associated group name allows the association of an IP address pool with a named group. Any IP address pool created *without* the **group** keyword automatically becomes a member of a base system group.

An IP address pool name can be associated with only one group. Subsequent use of the same pool name, within a pool group, is treated as an extension of that pool, and any attempt to associate an existing local IP address pool name with a different pool group is rejected. Therefore, each use of a pool name is an implicit selection of the associated pool group.

Note To reduce the chances of inadvertent generation of duplicate addresses, the system allows creation of the special pool named "default" only in the base system group; that is, no group name can be specified with the pool name "default."

All IP address pools within a pool group are checked to prevent overlapping addresses; however, no checks are made between any group pool member and a pool not in a group. The specification of a named pool within a pool group allows the existence of overlapping IP addresses with pools in other groups, and with pools in the base system group, but not among pools within a group. Otherwise, processing of the IP address pools is not altered by their membership in a group. In particular, these pool names can be specified in **peer** commands and returned in RADIUS and AAA functions with no special processing.

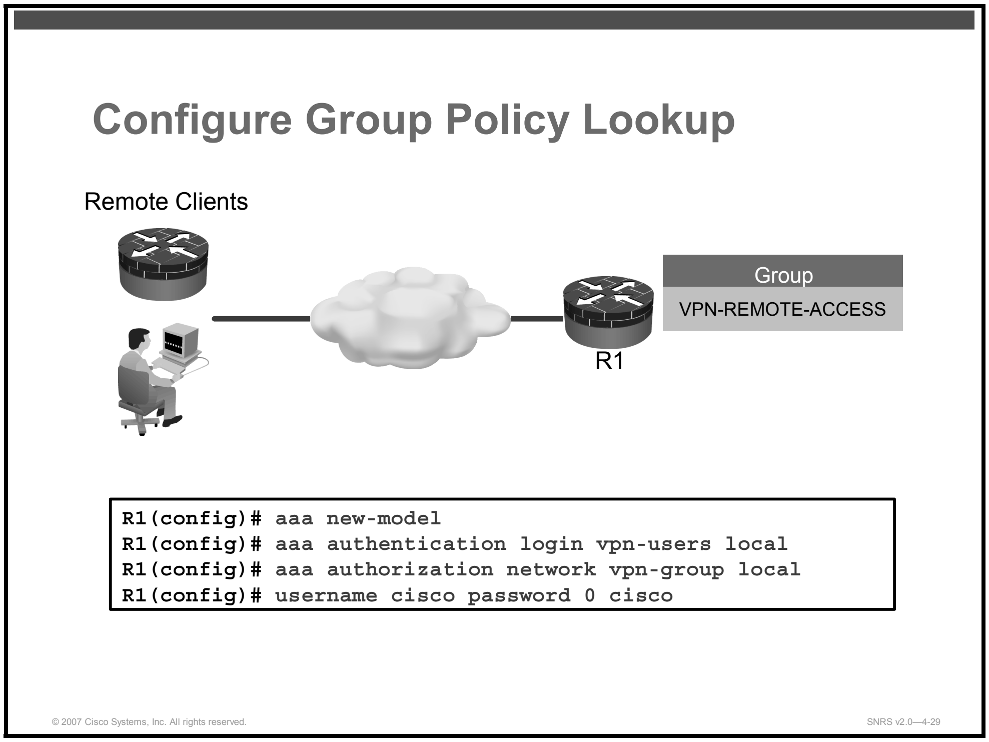
IP address pools can be associated with VPNs. This association permits flexible IP address pool specifications that are compatible with a VPN and a VPN routing and forwarding (VRF) instance.

The IP address pools can also be used with the **translate** commands for one-step vty-async connections and in certain AAA or TACACS+ authorization functions.

IP address pools are displayed with the **show ip local pool** command in EXEC mode.

Enable Group Policy Lookup via AAA

This section describes how to configure group policy lookup.



Configure Group Policy Lookup

```
R1(config)# aaa new-model
R1(config)# aaa authentication login vpn-users local
R1(config)# aaa authorization network vpn-group local
R1(config)# username cisco password 0 cisco
```

SNRS v2.0—4-29

Configuring group policy lookup is completed in two steps, as shown in the figure and listed here:

Step 1 Enable AAA.

```
R1(config)# aaa new-model
```

Step 2 Set AAA authentication at login.

```
R1(config)# aaa authentication login {default | list-name}
password-expiry method1 [method2...]
```

Syntax Description

default	Uses the listed authentication methods that follow this keyword as the default list of methods when a user logs in
list-name	Character string used to name the list of authentication methods activated when a user logs in
password-expiry	Enables password aging on a local authentication list
method1 [*method2...*]	Identifies the list of methods that the authentication algorithm tries in the given sequence You must enter at least one method; you may enter up to four methods.

If the **default** list is not set, only the local user database is checked.

Note This command must be enabled to enforce XAUTH.

Step 3 Enable group policy lookup using the **aaa authorization network** command. A RADIUS server and the router local database may be used together and are tried in the order listed.

```
R1(config)# aaa authorization {network | exec | commands level
| reverse-access | configuration} {default | list-name}
[method1 [method2...]]
```

Syntax Description

network	Runs authorization for all network-related service requests, including Serial Line Internet Protocol (SLIP), PPP, PPP Network Control Protocol (NCP), and AppleTalk Remote Access (ARA)
exec	Runs authorization to determine if the user is allowed to run an EXEC shell This facility might return user profile information such as autocommand information.
commands	Runs authorization for all commands at the specified privilege level
level	Specific command level that should be authorized Valid entries are 0 through 15.
reverse-access	Runs authorization for reverse access connections, such as reverse Telnet.
configuration	Downloads the configuration from the AAA server
default	Uses the listed authorization methods that follow this keyword as the default list of methods for authorization
list-name	Character string used to name the list of authorization methods
method1 [*method2...*]	Identifies an authorization method or multiple authorization methods to be used for authorization

Use the **aaa authorization** command to enable authorization and to create named methods lists, defining authorization methods that can be used when a user accesses the specified function. Method lists for authorization define the ways that authorization will be performed and the sequence in which these methods will be performed. A method list is a named list describing the authorization methods to be used (such as RADIUS or TACACS+), in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Cisco IOS Software uses the first method listed to authorize users for specific network services; if that method fails to respond, the Cisco IOS Software selects the next method listed in the method list. This process continues until there is successful communication with a listed authorization method or until all methods defined are exhausted.

Note The Cisco IOS Software attempts authorization with the next listed method only when there is no response from the previous method. If authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user services—the authorization process stops and no other authorization methods are attempted.

If the **aaa authorization** command for a particular authorization type is issued without a named method list specified, the default method list is automatically applied to all interfaces or lines (where this authorization type applies) except those that have a named method list explicitly defined. (A defined method list overrides the default method list.) If no default method list is defined, no authorization takes place.

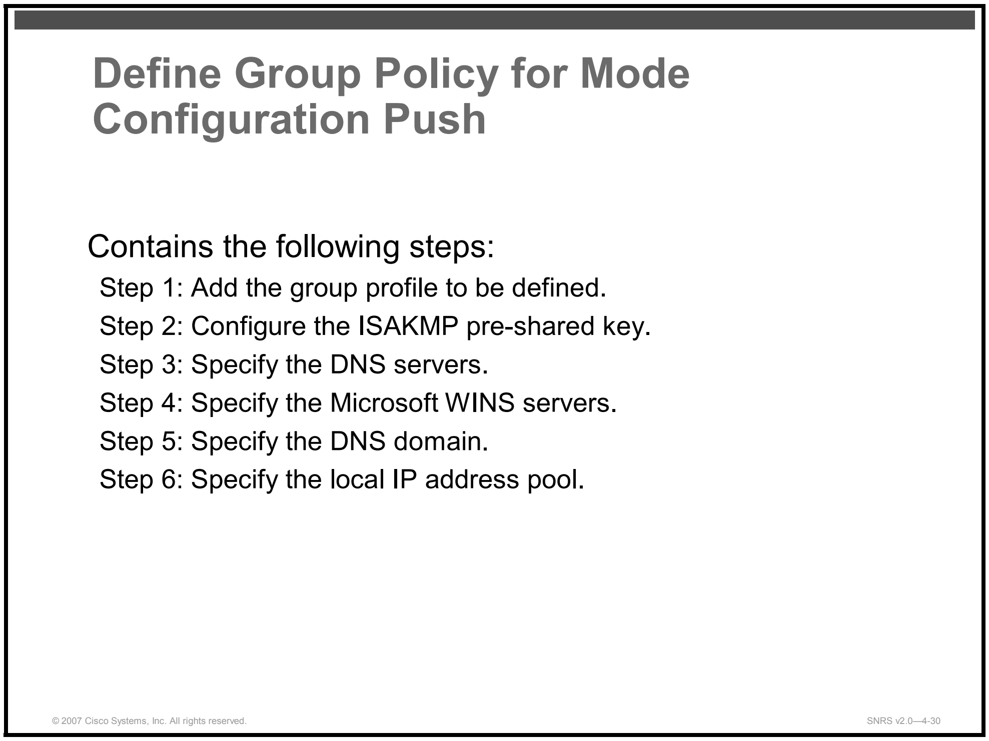
Use the **aaa authorization** command to create a list by entering values for the *list-name* and the *method* arguments, where *list-name* is any character string used to name this list (excluding all method names) and *method* identifies the list of authorization methods tried in the given sequence.

Step 4 (Optional) Define local users for XAUTH if RADIUS or TACACS+ is not used.

```
R1(config)# username name password encryption-type encrypted-
password
```

Define Group Policy for Mode Configuration Push

This section describes the steps involved in defining the policy attributes that are pushed to the client via mode configuration.



Define Group Policy for Mode Configuration Push

Contains the following steps:

Step 1: Add the group profile to be defined.

Step 2: Configure the ISAKMP pre-shared key.

Step 3: Specify the DNS servers.

Step 4: Specify the Microsoft WINS servers.

Step 5: Specify the DNS domain.

Step 6: Specify the local IP address pool.

SNRS v2.0—4-30

Complete this task to define a group policy to be pushed during mode configuration. Although users can belong to only one group per connection, they may belong to specific groups with different policy requirements. Thus, users may decide to connect to the client using a different group ID by changing their client profile on the VPN device. To define the policy attributes that are pushed to the client via mode configuration, perform the steps listed in this section.

Complete the following steps beginning in global configuration mode to define the policy attributes that are pushed to the Cisco Easy VPN Remote client via mode configuration:

Step 1 Add the group profile to be defined.

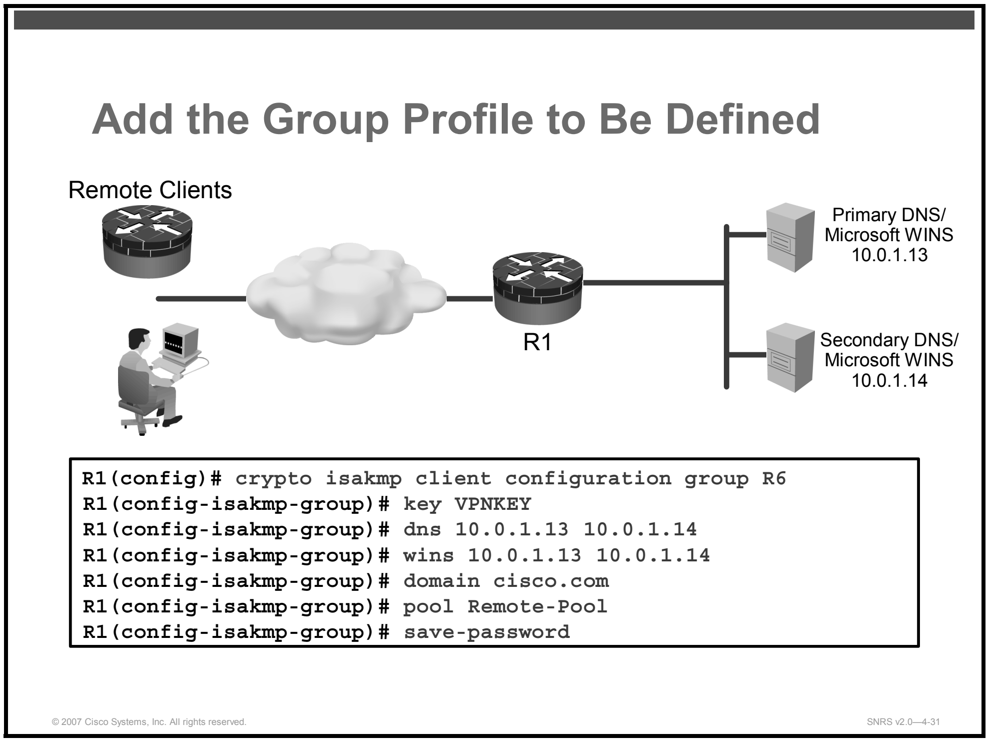
Step 2 Configure the ISAKMP pre-shared key.

Step 3 Specify the DNS servers.

Step 4 Specify the Microsoft Windows Internet Name Service (WINS) servers.

Step 5 Specify the DNS domain.

Step 6 Specify the local IP address pool.



Add the Group Profile to Be Defined

```
R1(config)# crypto isakmp client configuration group R6
R1(config-isakmp-group)# key VPNKEY
R1(config-isakmp-group)# dns 10.0.1.13 10.0.1.14
R1(config-isakmp-group)# wins 10.0.1.13 10.0.1.14
R1(config-isakmp-group)# domain cisco.com
R1(config-isakmp-group)# pool Remote-Pool
R1(config-isakmp-group)# save-password
```

SNRS v2.0—4-31

The **crypto isakmp client configuration group** command specifies the policy profile of the group that will be defined and enters ISAKMP group configuration mode.

Complete these steps to configure the profile for a defined group.

Step 1 Specify group policy information that needs to be defined or changed.

R1(config)# **crypto isakmp client configuration group** {*group-name* | **default**}

Syntax Description

group-name	Group definition that identifies which policy is enforced for users
default	Policy that is enforced for all users who do not offer a group name that matches a *group-name* argument The default keyword can only be configured locally.

Use the **crypto isakmp client configuration group** command to specify group policy information that needs to be defined or changed. You may wish to change the group policy on your router if you decide to connect to the client using a group ID that does not match the *group-name* argument.

After enabling this command, which puts you in ISAKMP group configuration mode, you can specify characteristics for the group policy using the following commands:

- **access-restrict**: Ties a particular VPN group to a specific interface for access to the Cisco IOS gateway and the services that it protects
- **acl:** Configures split tunneling
- **auto-update-client**: Configures auto upgrade

- **backup-gateway:** Configures a server to “push down” a list of backup gateways to the client (These gateways are tried in order in the case of a failure of the previous gateway. The gateways may be specified using IP addresses or hostnames.)
- **banner:** Specifies a mode configuration banner
- **browser-proxy:** Applies a browser proxy map to a group
- **configuration url:** Specifies on a server the URL that a Cisco Easy VPN Remote device must use to get a configuration in a mode configuration exchange
- **configuration version:** Specifies on a server the version that a Cisco Easy VPN Remote device must use to get a particular configuration in a mode configuration exchange
- **crypto aaa attribute list:** Defines a AAA attribute list of per-user attributes on a local Cisco Easy VPN Server
- **dhcp server:** Configures multiple DHCP server entries
- **dhcp timeout:** Controls the wait time before the next DHCP server on the list is tried
- **dns:** Specifies the primary and secondary DNS servers for the group
- **domain:** Specifies group domain membership
- **firewall are-u-there:** Adds the Firewall-Are-U-There attribute to the server group if your PC is running the Black Ice or Zone Alarm personal firewalls
- **firewall policy:** Specifies the Centralized Protection Policy (CPP) firewall policy push name for the crypto ISAKMP client configuration group on a local AAA server
- **group-lock:** Use if pre-shared key authentication is used with IKE; allows you to enter your XAUTH username (The group delimiter is compared against the group identifier sent during IKE aggressive mode.)
- **include-local-lan:** Configures the Include-Local-LAN attribute to allow a nonsplit tunneling connection to access the local subnetwork at the same time as the client
- **key:** Specifies the IKE pre-shared key when defining group policy information for mode configuration push
- **max-logins:** Limits the number of simultaneous logins for users in a specific user group
- **max-users:** Limits the number of connections to a specific server group
- **netmask:** Subnet mask to be used by the client for local connectivity
- **pfs:** Configures a server to notify the client of the central-site policy regarding whether PFS is required for any IPsec SA (Because the client device does not have a user interface option to enable or disable PFS negotiation, the server will notify the client device of the central-site policy via this parameter. The DH group that is proposed for PFS will be the same that was negotiated in Phase 1 of the IKE negotiation.)
- **Pool:** Refers to the IP local pool address used to allocate internal IP addresses to clients
- **save-password:** Saves your XAUTH password locally on your PC
- **split-dns:** Specifies a list of domain names that must be tunneled or resolved to the private network
- **wins:** Specifies the primary and secondary Microsoft WINS servers for the group

Step 2 Specify the IKE pre-shared key for group policy attribute definition.

```
R1(config-isakmp-group)# key string
```

Syntax Description

name	Name of the shared key

Note This command must be enabled if the client identifies itself with a pre-shared key.

Step 3 (Optional) Specify the primary and secondary DNS servers.

```
R1(config-isakmp-group)# dns primary-server secondary-server
```

Syntax Description

primary-server	Name or IP address of the primary DNS server
secondary-server	Name or IP address of the secondary DNS server

You can also specify by hostname.

Step 4 (Optional) Specify the primary and secondary Microsoft WINS servers.

```
R1(config-isakmp-group)# wins primary-server secondary-server
```

Syntax Description

primary-server	Name or IP address of the primary Microsoft WINS server
secondary-server	Name or IP address of the secondary Microsoft WINS server

Step 5 (Optional) Specify the DNS domain to which a group belongs.

```
R1(config-isakmp-group)# domain name
```

Syntax Description

name	Name of the DNS domain

Step 6 Use the **pool** command to refer to a local address pool, which defines a range of addresses that will be used to allocate an internal IP address to a Cisco Easy VPN Remote client,

```
R1(config-isakmp-group)# pool name
```

Syntax Description

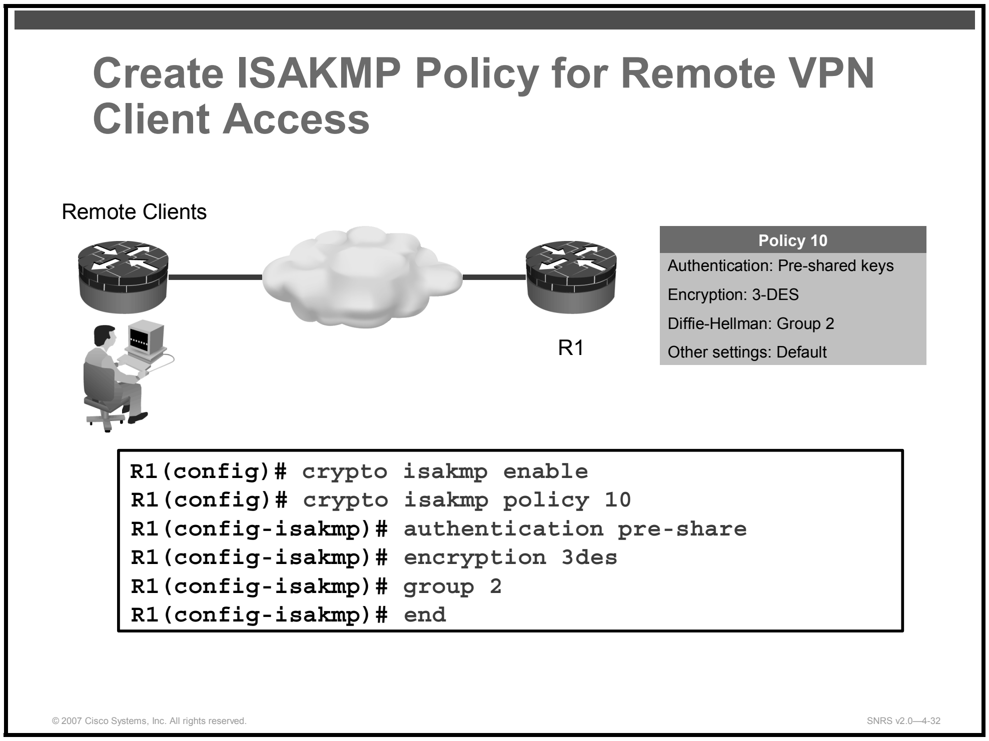
name	Name of the local pool

Step 7 (Optional) Save your XAUTH password locally.

```
R1(config-isakmp-group)# save-password
```

Create ISAKMP Policy for Cisco Easy VPN Remote Client Access

This section describes the commands used to create your ISAKMP policies.



Create ISAKMP Policy for Remote VPN Client Access

Policy 10
Authentication: Pre-shared keys
Encryption: 3-DES
Diffie-Hellman: Group 2
Other settings: Default

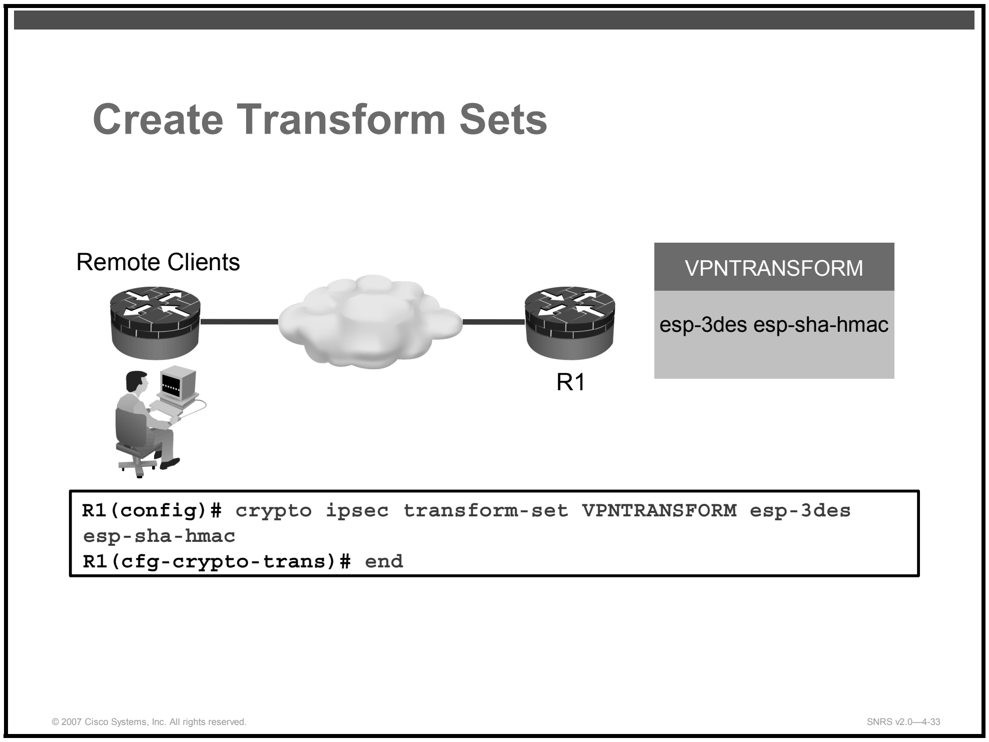
```
R1(config)# crypto isakmp enable
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encryption 3des
R1(config-isakmp)# group 2
R1(config-isakmp)# end
```

SNRS v2.0—4-32

Complete this task to configure the ISAKMP policy for all Cisco Easy VPN Remote clients attaching to this router. Use the standard ISAKMP configuration commands to accomplish this task. The figure shows an example of how to configure the ISAKMP policy.

Create a Transform Set

This section describes how to create a transform set to be exchanged with clients.



Create Transform Sets

```
R1(config)# crypto ipsec transform-set VPNTRANSFORM esp-3des
esp-sha-hmac
R1(cfg-crypto-trans)# end
```

SNRS v2.0—4-33

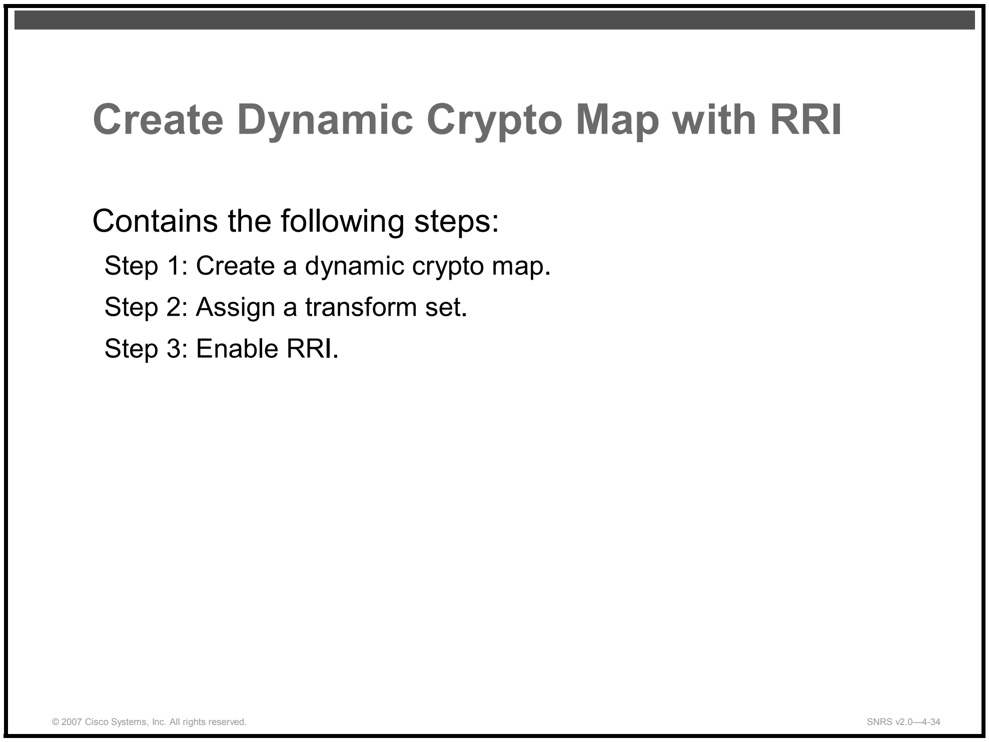
This task creates a transform set for the Cisco Easy VPN Remote clients to use when they attempt to build an IPsec tunnel to this router. Use the standard method for creating a transform set, as shown in this figure.

Here is an example of how to create a transform set for Cisco Easy VPN Remote client access:

```
R1(config)# crypto ipsec transform-set transform-set-name transform1
[transform2 [transform3]]
```

Create a Dynamic Crypto Map with RRI

This section describes how to enable RRI for the Cisco Easy VPN Remote client.



Create Dynamic Crypto Map with RRI

Contains the following steps:

Step 1: Create a dynamic crypto map.

Step 2: Assign a transform set.

Step 3: Enable RRI.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-34

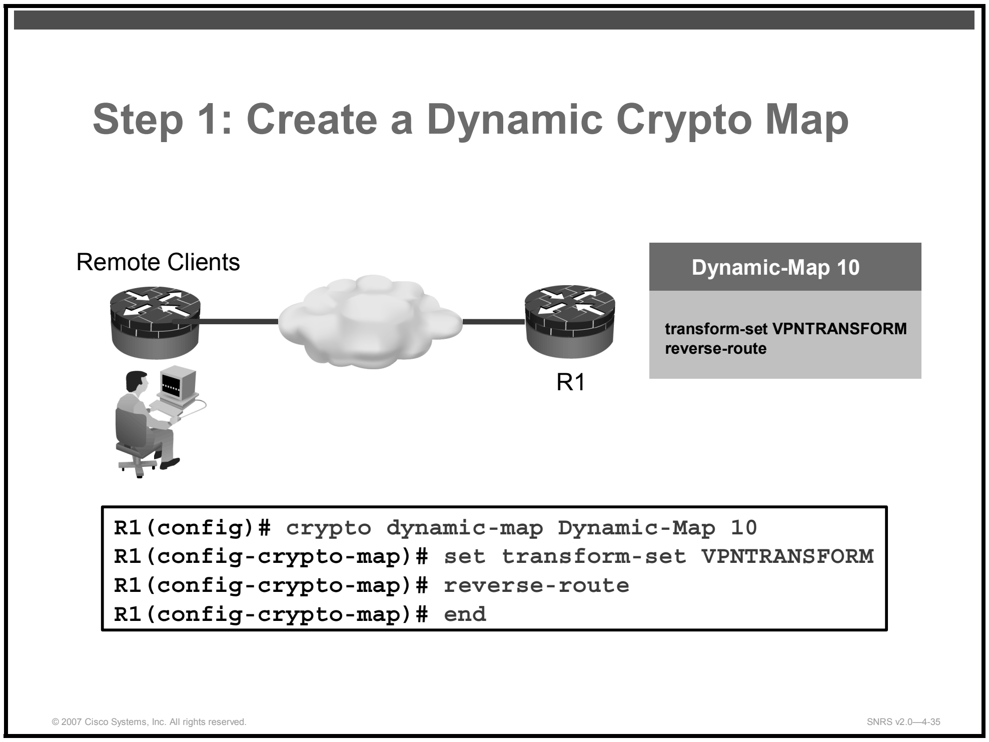
This task creates a dynamic crypto map to be used when building IPsec tunnels to Cisco Easy VPN Remote clients. In this example, RRI is used to ensure that returning data destined for a particular IPsec tunnel can find that tunnel. RRI ensures that a static route is created on the Cisco Easy VPN Server for each client internal IP address.

Complete the following steps to create the dynamic crypto map with RRI:

Step 1 Create a dynamic crypto map.

Step 2 Assign a transform set to the crypto map.

Step 3 Enable RRI.



Step 1: Create a Dynamic Crypto Map

```
R1(config)# crypto dynamic-map Dynamic-Map 10
R1(config-crypto-map)# set transform-set VPNTRANSFORM
R1(config-crypto-map)# reverse-route
R1(config-crypto-map)# end
```

SNRS v2.0—4-35

Complete these steps to create a dynamic crypto map.

Step 1 Create a dynamic crypto map entry and enter the crypto map configuration mode using the **crypto dynamic-map** command.

```
R1(config)# crypto dynamic map-name seq-num
```

Syntax Description

dynamic-map-name	Specifies the name of the dynamic crypto map set
dynamic-seq-num	Specifies the number of the dynamic crypto map entry

A dynamic crypto map entry is essentially a crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPsec negotiation) to match the requirements of a remote peer. This practice allows remote peers to exchange IPsec traffic with the router even if the router does not have a crypto map entry specifically configured to meet all of the requirements of the remote peer.

Dynamic crypto maps have these characteristics:

- They are not used by the router to initiate new IPsec SAs with remote peers.
- They are used when a remote peer tries to initiate an IPsec SA with the router.
- They are used in evaluating traffic.

Step 2 Specify which transform sets are allowed for the crypto map entry. When using this command, be sure to list multiple transform sets in order of priority (highest priority first).

```
R1(config-crypto-map)# set transform-set transform-set-name
[transform-set-name2…transform-set-name6]
```

Note This is the only configuration statement required in dynamic crypto map entries.

Syntax Description

transform-set-name	Name of the transform set: ■ For an IPsec manual crypto map entry, you can specify only one transform set. ■ For an IPsec ISAKMP or dynamic crypto map entry, you can specify up to six transform sets.

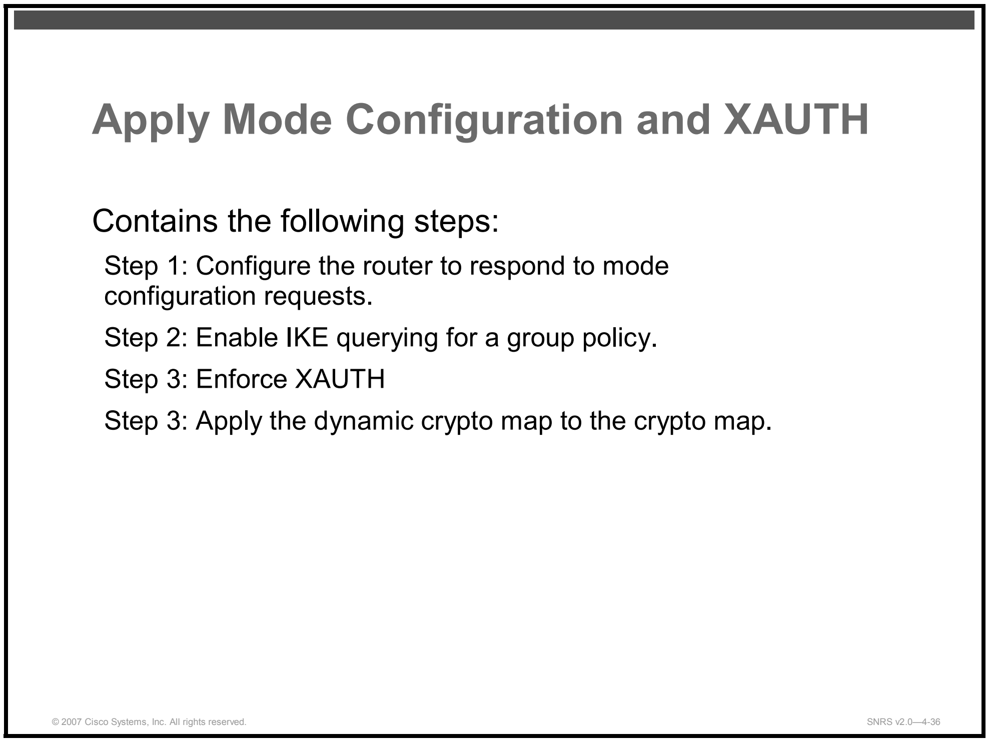
Step 3 Enable RRI using the **reverse-route** command.

```
R1(config-crypto-map)# reverse-route
```

This command has no arguments or keywords.

Apply Mode Configuration to the Crypto Map

This section describes how apply mode configuration to the dynamic crypto map.



Apply Mode Configuration and XAUTH

Contains the following steps:

Step 1: Configure the router to respond to mode configuration requests.

Step 2: Enable IKE querying for a group policy.

Step 3: Enforce XAUTH

Step 3: Apply the dynamic crypto map to the crypto map.

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-36

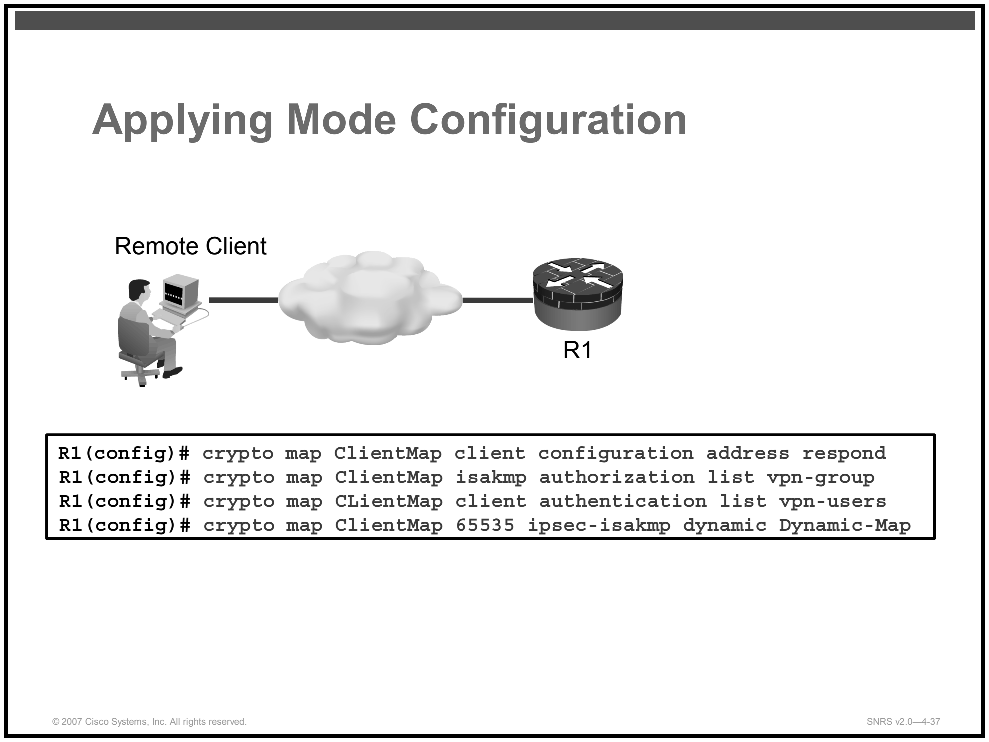
Apply mode configuration to a dynamic crypto map using the following steps in global configuration mode:

Step 1 Configure the router to respond to mode configuration requests.

Step 2 Enable IKE queries for group policy lookup.

Step 3 Enforce Xauth (Optional)

Step 4 Apply the dynamic crypto map to the crypto map.



Applying Mode Configuration

```
R1(config)# crypto map ClientMap client configuration address respond
R1(config)# crypto map ClientMap isakmp authorization list vpn-group
R1(config)# crypto map CLientMap client authentication list vpn-users
R1(config)# crypto map ClientMap 65535 ipsec-isakmp dynamic Dynamic-Map
```

SNRS v2.0—4-37

Complete these steps to apply mode configuration.

Step 1 Configure the router to initiate or reply to mode configuration requests.

```
R1(config)# crypto map map-name client configuration address
[initiate | respond]
```

Note Cisco clients require the **respond** keyword to be used; however, if the Cisco VPN Client v1.x is used, the **initiate** keyword must be used; the **initiate** and **respond** keywords may be used simultaneously.

Syntax Description

map-name	The name that identifies the crypto map
initiate	Pushes the network address to the client
respond	A keyword that indicates that the router will accept requests for IP addresses from any requesting peer

Step 2 Enable IKE querying for a group policy when requested by the Cisco VPN Client. AAA uses the *list-name* argument to determine which method list is used to find the policy (local or RADIUS) as defined in the **aaa authorization network** command.

```
R1(config)# crypto map map-name isakmp authorization list
list-name
```

Syntax Description

map-name	Name that you assign to the crypto map set
list-name	Character string used to name the list of authorization methods activated when a user logs in The list name must match the list name defined during AAA configuration.

Use this command to enable key lookup from a AAA server.

Pre-shared keys deployed in a large-scale VPN without a certification authority, with dynamic IP addresses, are accessed during aggression mode of IKE negotiation through a AAA server. Thus, users have their own key, which is stored on an external AAA server. This allows for central management of the user database, linking it to an existing database, in addition to allowing all users to have their own unique, more secure pre-shared key.

Before configuring the **crypto map client authorization list** command, you should perform the following tasks:

- Set up an authorization list using AAA commands
- Configure an IPsec transform
- Configure a crypto map
- Configure an ISAKMP policy using IPsec and IKE commands

After enabling the **crypto map client authorization list** command, you should apply the previously defined crypto map to the interface.

Step 3 (Optional) Enforce XAUTH.

```
R1(config)# crypto map map-name client authentication list
list-name
```

Syntax Description

map-name	The name you assign to the crypto map set
list-name	Character string used to name the list of authentication methods activated when a user logs in The *list-name* argument must match the list name defined during AAA configuration.

Before configuring XAUTH, you should complete the following tasks:

- Set up an authentication list using AAA commands
- Configure an IPsec transform
- Configure a crypto map
- Configure an ISAKMP policy

After enabling XAUTH, you should apply the crypto map on which XAUTH is configured to the router interface.

Step 4 Apply the dynamic crypto map to the crypto map.

```
R1(config)# crypto map map-name seq-number ipsec-isakmp
dynamic dynamic-map-name
```

The syntax for the **crypto map** command is as follows:

```
crypto map map-name seq-num ipsec-manual
```

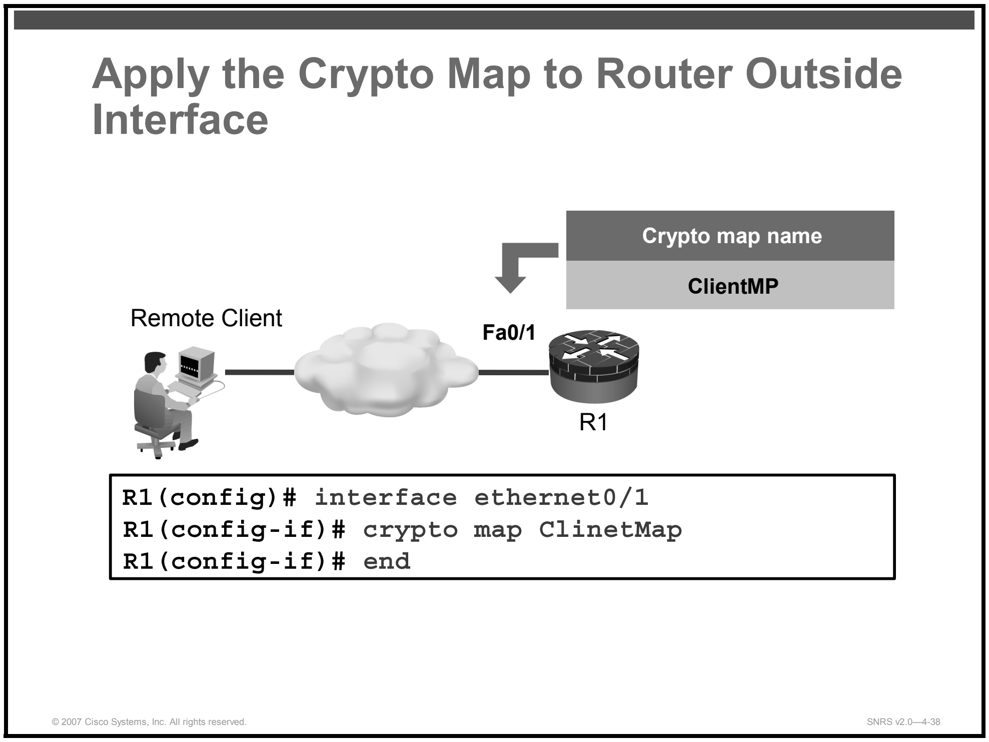
```
crypto map map-name seq-num ipsec-isakmp [dynamic dynamic-map-name]
[discover]
```

Syntax Description

map-name	The name you assign to the crypto map set
seq-num	The number you assign to the crypto map entry
ipsec-manual	Indicates that IKE will not be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry
ipsec-isakmp	Indicates that IKE will be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry
dynamic	(Optional) Specifies that this crypto map entry is to reference a pre-existing dynamic crypto map Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPsec device. If you use this keyword, none of the crypto map configuration commands will be available.
dynamic-map-name	(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template
discover	(Optional) Enables peer discovery; by default, peer discovery is not enabled.

Apply the Crypto Map to the Router Interface

This section describes the command used to apply the crypto map to a router interface.



Apply the Crypto Map to Router Outside Interface

```
R1(config)# interface ethernet0/1
R1(config-if)# crypto map ClinetMap
R1(config-if)# end
```

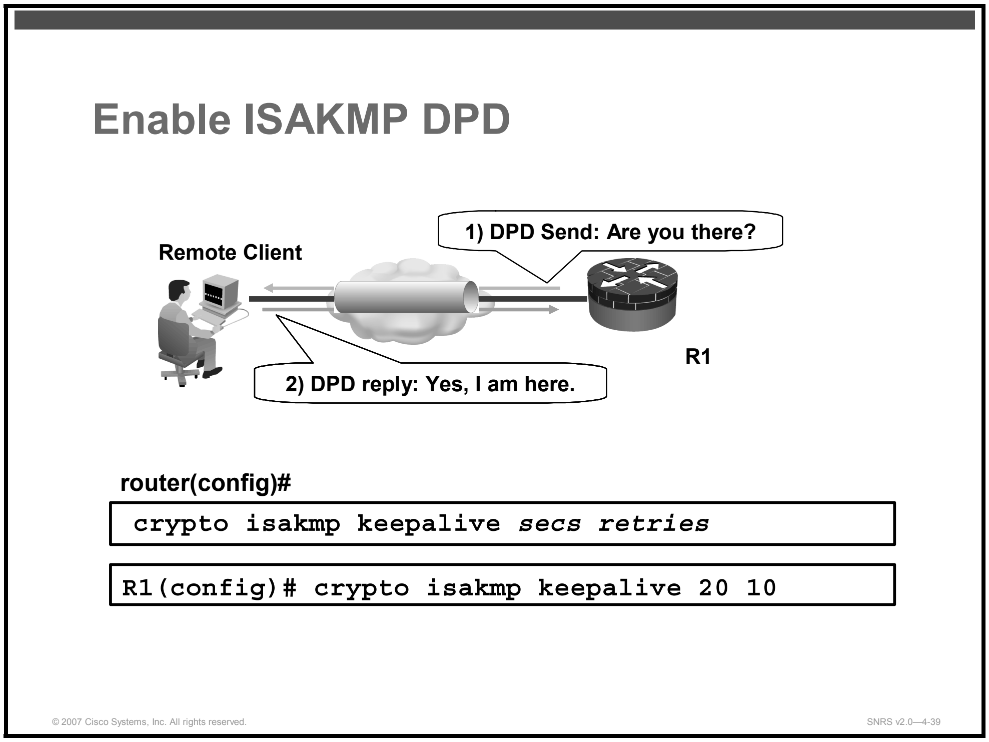
SNRS v2.0—4-38

This task applies the crypto map to the Cisco Easy VPN Server router outside interface.

The figure above shows an example of how to apply the crypto map to the outside interface.

Enable ISAKMP DPD (Optional)

This section describes how to enable DPD.



Enable ISAKMP DPD

router(config)#

```
crypto isakmp keepalive secs retries
```

```
R1(config)# crypto isakmp keepalive 20 10
```

Use the **crypto isakmp keepalive** command in global configuration mode to enable a Cisco IOS VPN gateway (instead of the Cisco VPN Client) to send ISAKMP DPD messages.

The syntax for the **crypto isakmp keepalive** command is as follows:

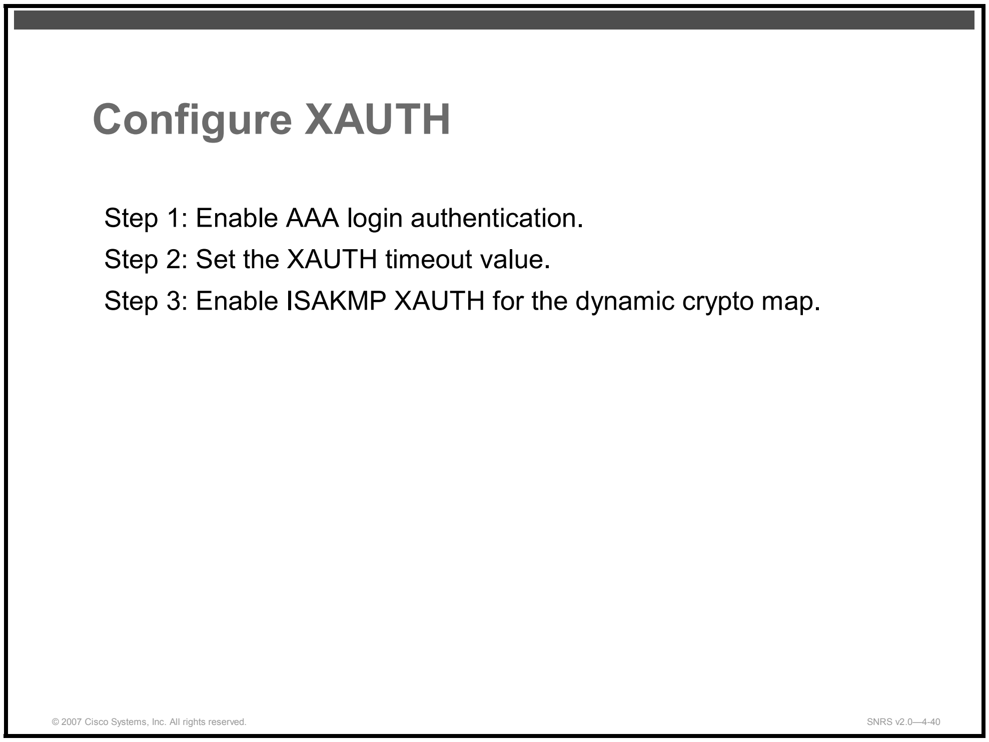
```
        crypto isakmp keepalive secs retries
```

Syntax Description

secs	Specifies the number of seconds between DPD messages; the range is between 10 and 3600 seconds.
retries	Specifies the number of seconds between retries if DPD messages fail; the range is between 2 and 60 seconds.

Configure or Disable XAUTH

This section describes how to configure Extended Authentication (XAUTH).



Configure XAUTH

Step 1: Enable AAA login authentication.

Step 2: Set the XAUTH timeout value.

Step 3: Enable ISAKMP XAUTH for the dynamic crypto map.

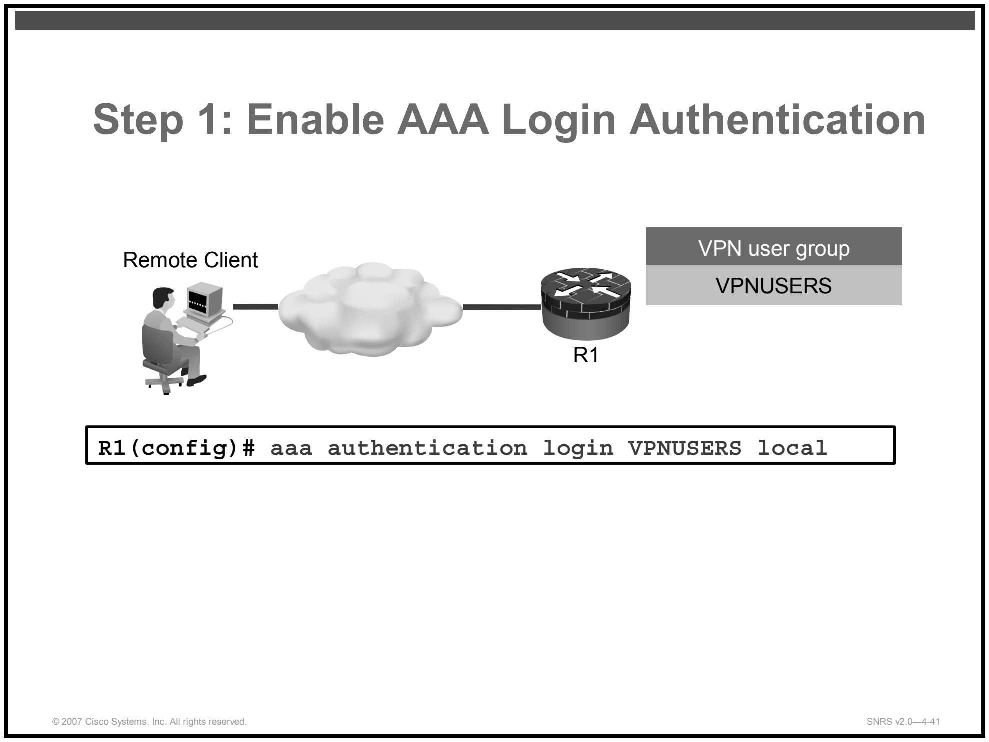
SNRS v2.0—4-40

Complete the following steps to configure XAUTH on your Cisco Easy VPN Server router:

Step 1 Enable AAA login authentication.

Step 2 Set the XAUTH timeout value.

Step 3 Enable ISAKMP XAUTH for the dynamic crypto map.



Step 1: Enable AAA Login Authentication

```
R1(config)# aaa authentication login VPNUSERS local
```

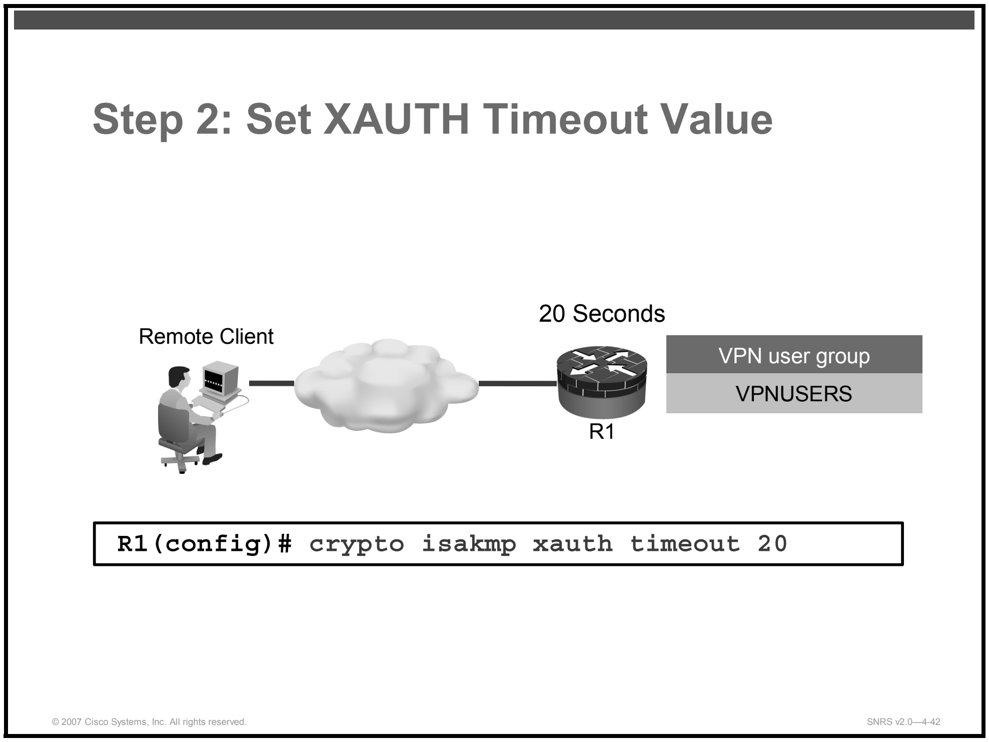
SNRS v2.0—4-41

Step 1 Enable AAA login authentication using the **aaa authentication login** command in global configuration mode.

The syntax for the **aaa authentication login** command is as follows:

aaa authentication login *list-name method1* [*method2…*]

Command Parameter	Description
list-name	Character string used to name the list of authentication methods activated when a user logs in The list name must match the list name defined during AAA configuration.
method	Keyword used to describe the authentication method used



Step 2: Set XAUTH Timeout Value

```
R1(config)# crypto isakmp xauth timeout 20
```

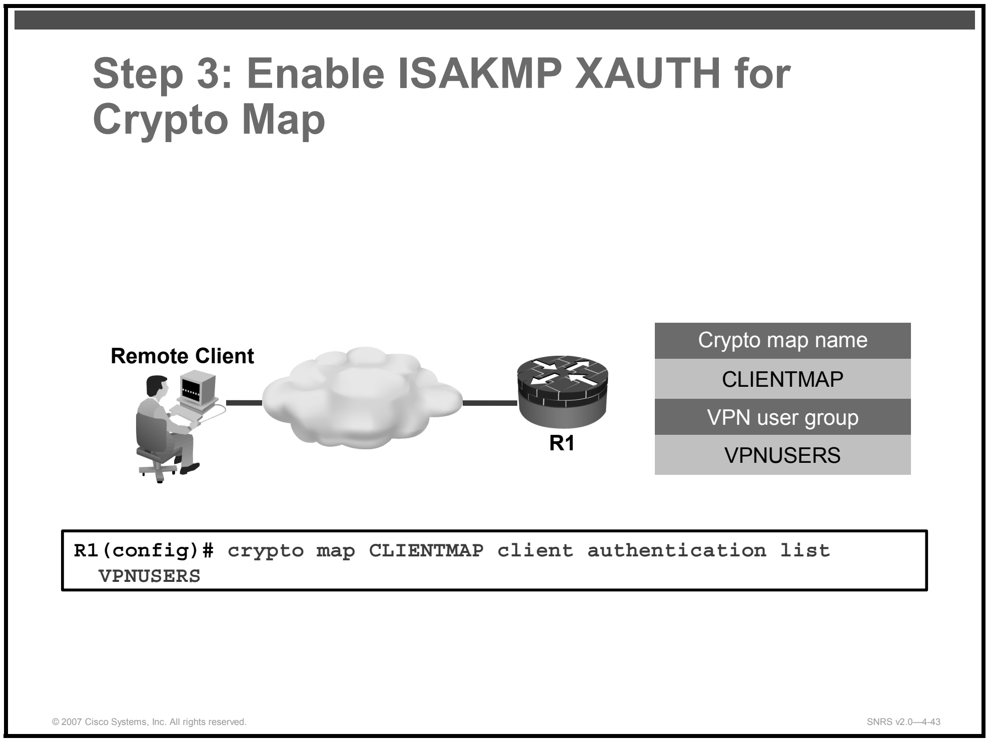
SNRS v2.0—4-42

Step 2 Set the XAUTH timeout value using the **crypto isakmp xauth timeout** command.

The syntax for the **crypto isakmp xauth timeout** command is as follows:

`crypto isakmp xauth timeout` *`seconds`*

Command Parameter	Description
seconds	The XAUTH timeout value in seconds



Step 3: Enable ISAKMP XAUTH for Crypto Map

```
R1(config)# crypto map CLIENTMAP client authentication list
  VPNUSERS
```

Step 3 Enable ISAKMP XAUTH for the dynamic crypto map using the **crypto map** command.

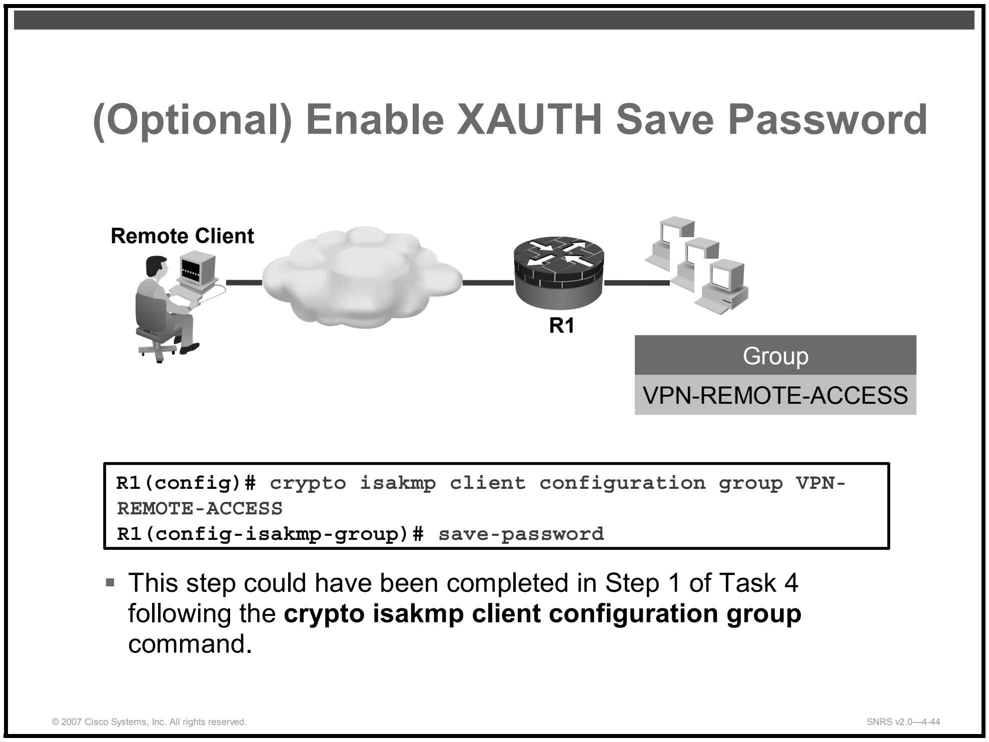
The syntax for the **crypto map** command is as follows:

`crypto map` *`map-name`* **`client authentication list`** *`list-name`*

Command Parameter	Description
map-name	Name that you assign to the crypto map set
list-name	Character string used to name the list of authentication methods activated when a user logs in The list name must match the list name defined during AAA configuration.

Enable XAUTH Save Password Feature

This section describes how to configure the optional XAUTH Save Password feature.



(Optional) Enable XAUTH Save Password

Group
VPN-REMOTE-ACCESS

```
R1(config)# crypto isakmp client configuration group VPN-
REMOTE-ACCESS
R1(config-isakmp-group)# save-password
```

- This step could have been completed in Step 1 of Task 4 following the **crypto isakmp client configuration group** command.

Cisco Easy VPN Remote uses one of three available authentication methods:

- **No XAUTH:** When no XAUTH is used, there is no authentication for the user when establishing the VPN tunnels. This is the least secure practice when configuring and using Cisco Easy VPN Remote.
- **XAUTH with no Save Password feature:** This is better than no XAUTH, but it requires that users re-enter the password each time that they need to establish the VPN tunnel (which may occur several times in one VPN session). Although this is the most secure form of authentication for Cisco Easy VPN Remote, it is also the most bothersome to users.
- **XAUTH with Save Password feature:** Using the Save Password function, users need only enter their passwords once when establishing the VPN tunnel. After that, the Cisco Easy VPN Remote automatically re-enters the password when required.

Enabling the XAUTH Save Password feature is an optional step. When configured, it allows the Cisco Easy VPN Remote client to save and reuse the last validated username and password for reauthentication. This means that a user no longer needs to re-enter the information manually. This step could have been done earlier while performing the **crypto isakmp client configuration group** command.

Use the **save-password** command in ISAKMP group configuration mode as shown in the figure.

The syntax for the **save-password** command is as follows:

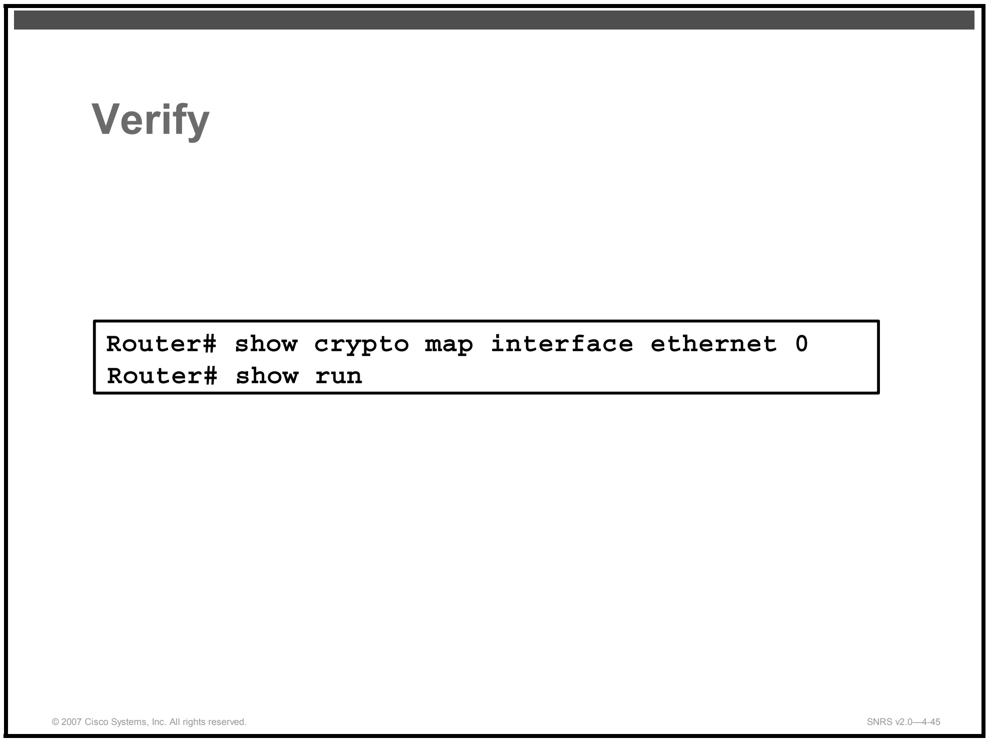
```
save-password
```

This command has no arguments or keywords.

Note The Save Password feature must be configured in both the Cisco Easy VPN Server and the Cisco Easy VPN Remote client.

Verify Easy VPN Server

This section describes how to verify your VPN server configuration.



Verify

```
Router# show crypto map interface ethernet 0
Router# show run
```

© 2007 Cisco Systems, Inc. All rights reserved.

SNRS v2.0—4-45

To verify your configurations for this feature, complete the following steps.

Step 1 Issue the **enable** command.

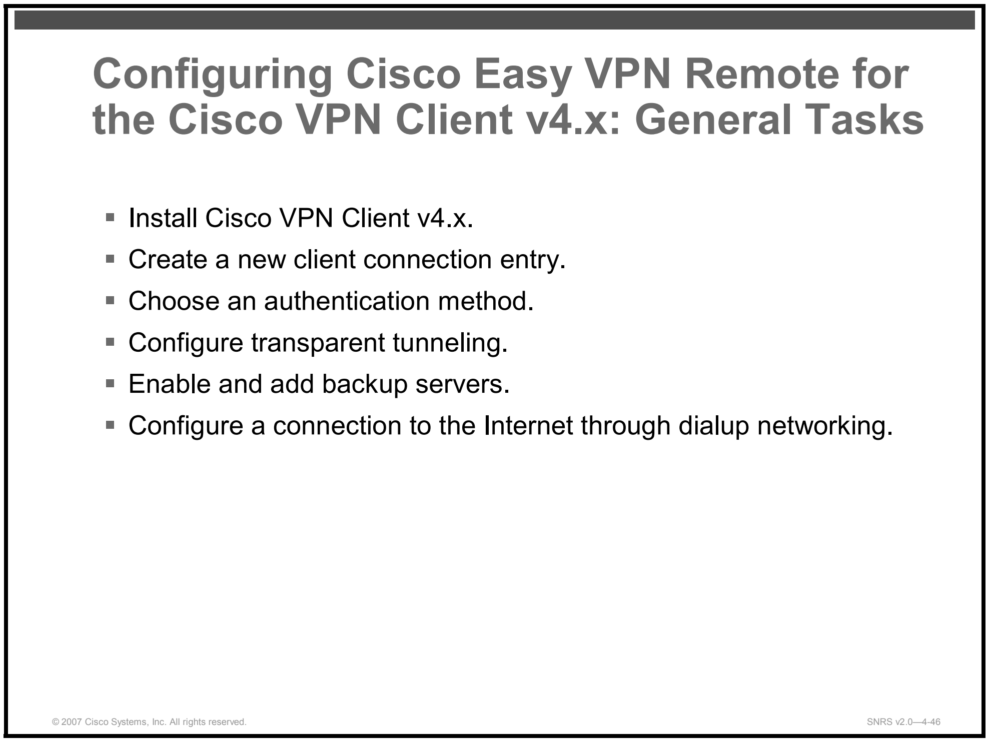
Step 2 Issue the **show crypto map** [**interface** *interface* | **tag** *map-name*] command.

Step 3 Issue the **show run** command.

	Command	Purpose
Step 1	**enable** Example: Router> **enable**	Enables privileged EXEC mode Enter your password if prompted.
Step 2	**show crypto map** [**interface** *interface* | **tag** *map-name*] Example: Router# **show crypto map interface ethernet 0**	Displays the crypto map configuration
Step 3	**show run**	Displays the running configuration

Configuring Cisco VPN Client v4.x

This topic describes how to configure Cisco VPN Client v4.x.



Configuring Cisco Easy VPN Remote for the Cisco VPN Client v4.x: General Tasks

- Install Cisco VPN Client v4.x.
- Create a new client connection entry.
- Choose an authentication method.
- Configure transparent tunneling.
- Enable and add backup servers.
- Configure a connection to the Internet through dialup networking.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-46

The Cisco VPN Client is simple to deploy and operate. The Cisco VPN Client enables customers to establish secure, end-to-end encrypted tunnels to any Cisco Easy VPN server. This thin design, IPSec implementation is available via Cisco.com for use with any Cisco central site remote access VPN product and is included free of charge with the Cisco VPN 3000 Concentrator, Cisco ASA 5500 Series security appliance, and most Cisco PIX Security Appliances.

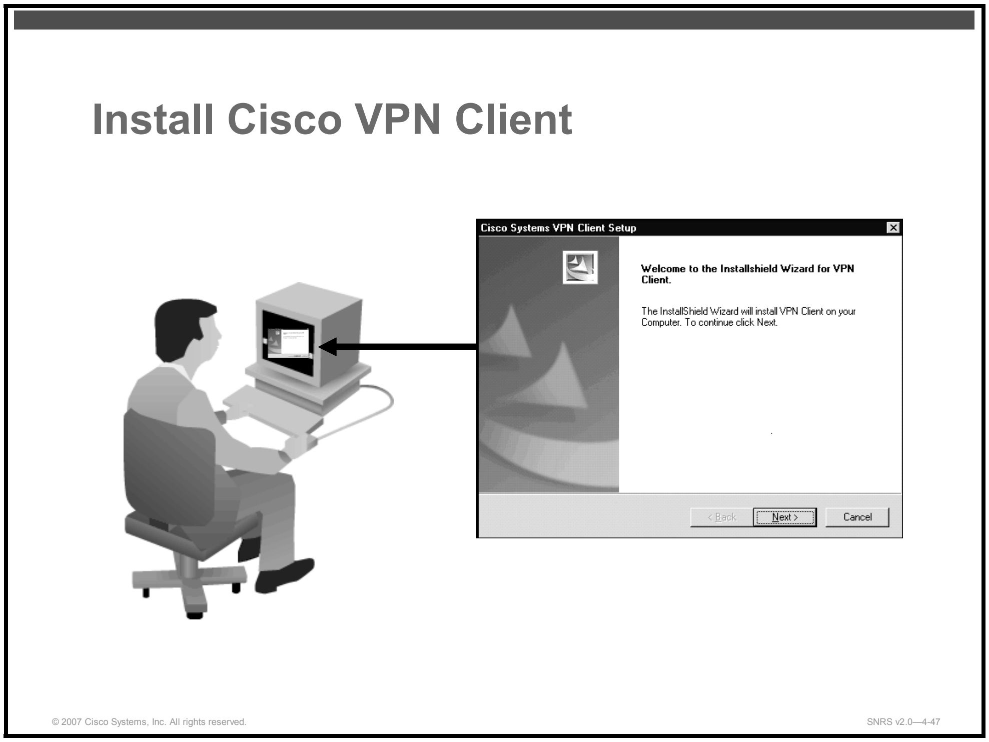
The client can be pre-configured for mass deployments and initial logins require very little user intervention. VPN access policies and configurations are downloaded from the central gateway and pushed to the client when a connection is established, allowing simple deployment and management, as well as high scalability. The Cisco VPN Client provides support for Windows 95(OSR2+), 98, ME, NT 4.0, 2000, XP, Linux (Intel), Solaris (UltraSparc-32 & 64 bit) and MAC OS X 10.1, 10.2 and 10.4.

Complete the following general tasks to configure the Cisco VPN Client for Cisco Easy VPN Remote access:

- **Task 1:** Install the Cisco VPN Client 4.x on the remote user PC.
- **Task 2:** Create a new client connection entry.
- **Task 3:** Choose an authentication method.
- **Task 4:** Configure transparent tunneling.
- **Task 5:** Enable and add backup servers.
- **Task 6:** Configure a connection to the Internet through dialup networking.

Install Cisco VPN Client

This section describes how to install a Cisco VPN Client on a computer.



Install Cisco VPN Client

Cisco Systems VPN Client Setup

Welcome to the Installshield Wizard for VPN Client.

The InstallShield Wizard will install VPN Client on your Computer. To continue click Next.

SNRS v2.0—4-47

You can install the Cisco VPN Client on your system through either of two applications: InstallShield and Microsoft Windows Installer. Both applications use installation wizards to walk you through the installation. Installing the Cisco VPN Client through InstallShield includes an uninstall icon in the program group; Windows Installer does not. In the latter case, to manually remove Cisco VPN Client applications, you can use the Microsoft Add/Remove Programs utility.

This topic explains how to install the Cisco VPN Client on your PC and includes the following:

- Verifying system requirements
- Gathering the information that you need
- Installing the Cisco VPN Client through InstallShield
- Installing the Cisco VPN Client through Windows Installer

Verifying System Requirements

Verify that your computer meets these requirements:

- A single, Pentium-class processor
- One of the following OSs:
  - Microsoft Windows 98 or Microsoft Windows 98 Second Edition
  - Microsoft Windows Me
  - Microsoft Windows NT 4.0 (with Service Pack 6 [SP6] or later)
  - Microsoft Windows 2000
  - Microsoft Windows XP
- Microsoft TCP/IP installed (To confirm, choose **Start > Settings > Control Panel > Network > Local Area Connection Properties**)
- 50 MB hard disk space
- RAM:
  - 32 MB for Microsoft Windows 98
  - 64 MB for Microsoft Windows NT and Microsoft Windows Me
  - 64 MB for Microsoft Windows 2000 (128 MB recommended)
  - 128 MB for Microsoft Windows XP (256 MB recommended)
- To install the Cisco VPN Client:
  - CD-ROM drive
  - 3.5-inch high-density diskette drive
  - Administrator privileges if installing on Microsoft Windows NT or Microsoft Windows 2000
- To use the Cisco VPN Client:
  - Direct network connection (cable or DSL modem and network adapter or interface card)
  - Internal or external modem
- To connect using a digital certificate for authentication:
  - A digital certificate signed by one of the following certificate authorities (CAs) installed on your PC:
    - Entrust (http://www.entrust.com)
    - Microsoft Certificate Services— Microsoft Windows 2000
    - Netscape Security
    - VeriSign (http://www.verisign.com)
  - Or a digital certificate stored on a smart card; the Cisco VPN Client supports smart cards via the Microsoft Cryptography application programming interface (CAPI)

Gathering the Information That You Need

To configure and use the Cisco VPN Client, you might need the information listed in this section.

Ask for this information from the system administrator of the private network that you want to access. Your system administrator might have preconfigured much of this data; if so, your system administrator will tell you which items you need.

- Hostname or IP address of the secure gateway to which you are connecting
- Your IPsec group name (for pre-shared keys)
- Your IPsec group password (for pre-shared keys)
- If authenticating with a digital certificate, the name of the certificate
- If authenticating through the internal server of the secure gateway, your username and password
- If authenticating through a RADIUS server, your username and password
- If authenticating through a Microsoft Windows NT domain server, your username and password
- If authenticating through a token vendor, your username and PIN
- If authenticating through a smart card, your smart card, reader, PIN or passcode, and the name of the certificate stored on the smart card
- If you should configure backup server connections, the hostnames or IP addresses of the backup servers

Installing the Cisco VPN Client Through InstallShield

To install the Cisco VPN Client on your system using InstallShield, follow these steps. It is suggested that you accept the defaults unless your system administrator has instructed you otherwise.

Step 1 Exit all Microsoft Windows programs, and disable any antivirus software.

Step 2 Insert the Cisco Systems CD-ROM in the CD-ROM drive of your system.

Step 3 Choose **Start > Run**. The Run dialog box appears.

Step 4 Enter **E:\VPN Client\CD-ROM\InstallShield\setup.exe**, where E: is the CD-ROM drive of your system.

Step 5 Click **OK**.

Note Cisco does not allow you to install the Cisco VPN Client software from a network drive. If you attempt to do so, you will receive an error message.

Step 6 If the InstallShield Wizard identifies an existing version of either the Cisco VPN 3000 Client or the Cisco VPN 5000 Client, it displays a dialog box that asks if you want to uninstall the existing client program. To continue, click **Yes**.

The Cisco VPN Client launches the appropriate uninstall wizard: the Cisco VPN Client uninstall wizard to uninstall a previous version of the Cisco VPN 3000 Client or the Cisco VPN

5000 Client. Follow the instructions on the uninstall wizard dialog boxes to automatically uninstall the program and reboot.

After your system reboots, the Cisco VPN Client Setup wizard resumes.

Step 7 Follow the instructions on the screens and enter a destination folder for the Cisco VPN Client files (or click **Next** to enter the default location C:\Program Files\Cisco Systems\VPN Client).

Step 8 You must restart your computer before you can configure and use the Cisco VPN Client.

Installing the Cisco VPN Client through Microsoft Windows Installer

Microsoft Windows Installer is available for Microsoft Windows NT, Microsoft Windows 2000, and Microsoft Windows XP.

Note If you are using Microsoft Windows Installer, you must have Microsoft Windows NT-based products such as Microsoft Windows NT 4.0 (with SP6), Microsoft Windows 2000, or Microsoft Windows XP. Installing with Microsoft Windows Installer also requires administrator privileges.

Microsoft Windows Installer 2.0 must be installed on a Microsoft Windows NT or Microsoft Windows 2000 PC before configuring the PC for a restricted user with elevated privileges.

To install the Cisco VPN Client using Microsoft Windows Installer, complete the following steps:

Step 1 Exit all Microsoft Windows programs, and disable any antivirus software.

Step 2 Insert the Cisco Systems CD-ROM in the CD-ROM drive of your system.

Step 3 Choose **Start > Run**. The Run dialog box appears.

Step 4 Enter **E:\VPN Client\CD-ROM\Msi\vpclient_en.exe**, where E: is the CD-ROM drive of your system.

Step 5 Click **OK**.

Note Cisco does not allow you to install the Cisco VPN Client software from a network drive. If you attempt to do so, you will receive an error message.

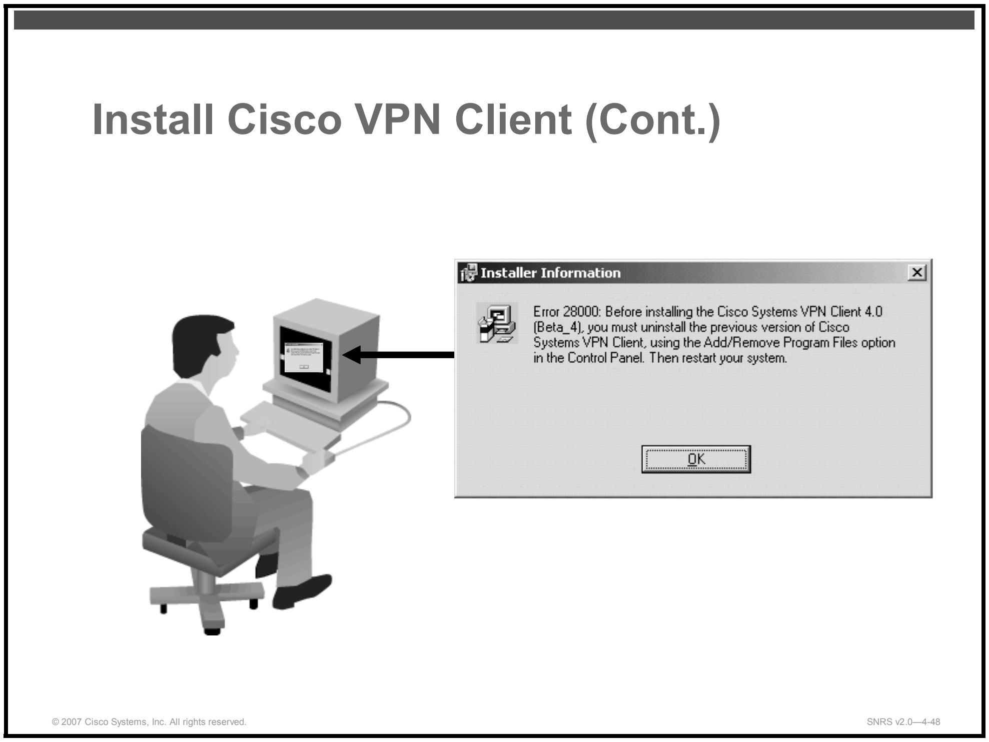
The program displays the Cisco Systems logo and the Microsoft Installer Setup window. Click **Next** to start the installation and then follow the instructions on the dialog boxes.

Microsoft Windows Installer installs the Cisco VPN Client in the default location C:\Program Files\Cisco Systems\VPN Client. If you want a different destination folder for the Cisco VPN Client files, enter the alternative location when prompted to do so.

When the installation has completed, the installer displays a confirmation dialog box.

Step 6 Click **Finish**. Microsoft Windows Installer prompts you to restart your system.

Step 7 Click **Yes** to restart your system.



Install Cisco VPN Client (Cont.)

Installer Information

Error 28000: Before installing the Cisco Systems VPN Client 4.0 (Beta_4), you must uninstall the previous version of Cisco Systems VPN Client, using the Add/Remove Program Files option in the Control Panel. Then restart your system.

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-48

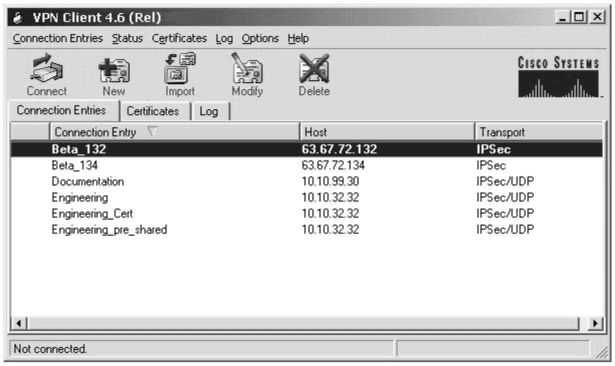
If you have not removed a previously installed Cisco VPN Client, when you execute the **vpnclient_en.exe** command or **vpnclient_en.msi** command, an error message displays. You must uninstall the previously installed Cisco VPN Client before proceeding with the new installation.

To remove a Cisco VPN Client installed with Microsoft Windows Installer, use the Microsoft Windows Add/Remove Programs control panel. To remove a Cisco VPN Client installed with InstallShield, choose **Start > Programs > Cisco Systems VPN Client > Uninstall Client**.

Create New Client Connection Entries

This section describes how to create a new client connection entry.

Create a New Client Connection Entry



Connection Entry	Host	Transport
Beta_132	63.67.72.132	IPSec
Beta_134	63.67.72.134	IPSec
Documentation	10.10.99.30	IPSec/UDP
Engineering	10.10.32.32	IPSec/UDP
Engineering_Cert	10.10.32.32	IPSec/UDP
Engineering_pre_shared	10.10.32.32	IPSec/UDP

Not connected.

To use the Cisco VPN Client, you must create at least one connection entry, which identifies the following information:

- The VPN device (the remote server) to access
- Pre-shared keys—the IPsec group to which the system administrator assigned you (Your group determines how you access and use the remote network. For example, it specifies access hours, number of simultaneous logins, user authentication method, and the IPsec algorithms that your Cisco VPN Client uses.)
- Certificates—the name of the certificate that you are using for authentication
- Optional parameters that govern Cisco VPN Client operation and connection to the remote network

You can create multiple connection entries if you use your Cisco VPN Client to connect to multiple networks (though not simultaneously) or if you belong to more than one VPN remote access group.

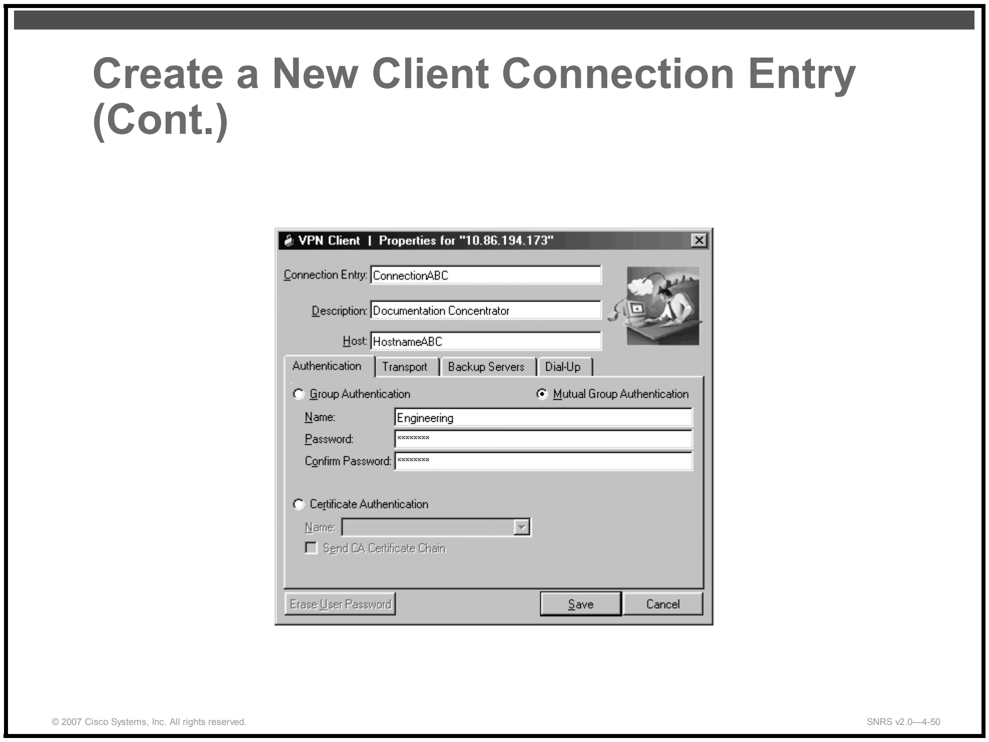
Creating a New Connection Entry

Use the following procedure to create a new connection entry.

Step 1 Start the Cisco VPN Client by choosing **Start > Programs > Cisco Systems VPN Client > VPN Client**.

Step 8 The Cisco VPN Client application starts and displays the advanced mode main window. If you are not already there, choose the **Options** menu in simple mode and choose **Advanced Mode** or press **Ctrl-M.**

Step 9 Choose **New** from the toolbar or the Connection Entries menu. The VPN Client displays a form.



Create a New Client Connection Entry (Cont.)

VPN Client | Properties for "10.86.194.173"

Connection Entry: ConnectionABC

Description: Documentation Concentrator

Host: HostnameABC

○ Group Authentication ● Mutual Group Authentication

Name: Engineering

Password: *********

Confirm Password: *********

○ Certificate Authentication

Name:

□ Send CA Certificate Chain

SNRS v2.0—4-50

Step 10 Enter a unique name for this new connection. You can use any name to identify this connection; for example, Engineering. This name can contain spaces, and it is not case-sensitive.

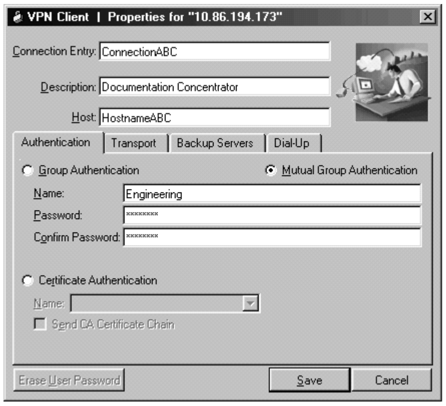
Step 11 Enter a description of this connection. This field is optional, but it helps further identify this connection; for example, Connection to Engineering remote server.

Step 12 Enter the hostname or IP address of the remote VPN device that you want to access.

Configure Client Authentication Properties

This section describes how to configure properties used during the client authentication process.

Configure Client Authentication Properties



Connection Entry: ConnectionABC

Description: Documentation Concentrator

Host: HostnameABC

○ Group Authentication ● Mutual Group Authentication

Name: Engineering

Password: *********

Confirm Password: *********

○ Certificate Authentication

Name:

□ Send CA Certificate Chain

© 2007 Cisco Systems, Inc. All rights reserved. SNRS v2.0—4-51

Under the Authentication tab, enter the information for the method that you want to use. You can connect as part of a group (configured on a VPN device) or by supplying an identity digital certificate.

Group Authentication

The network administrator usually configures group authentication for you. If this is not the case, complete the following procedure:

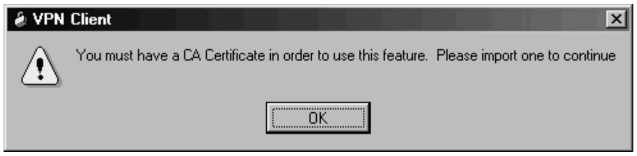
Step 1 Click the **Group Authentication** radio button.

Step 13 In the Name field, enter the name of the IPsec group to which you belong. This entry is case-sensitive.

Step 14 In the Password field, enter the password (which is also case-sensitive) for your IPsec group. The field displays only asterisks.

Step 15 Verify your password by entering it again in the Confirm Password field.

Mutual Group Authentication



VPN Client

You must have a CA Certificate in order to use this feature. Please import one to continue

Mutual Group Authentication

To use mutual group authentication, you need a root certificate that is compatible with the central-site VPN installed on your system. Your network administrator can load a root certificate on your system during installation. When you select mutual group authentication, the Cisco VPN Client software verifies whether you have a root certificate installed. If not, it prompts you to install one. Before you continue, you must import a root certificate.

When you have installed a root certificate (if required), follow the steps for group authentication.

Certificate Authentication

For certificate authentication, perform the following procedure, which varies according to the type of certificate that you are using:

Step 1 Click the Certificate Authentication radio button.

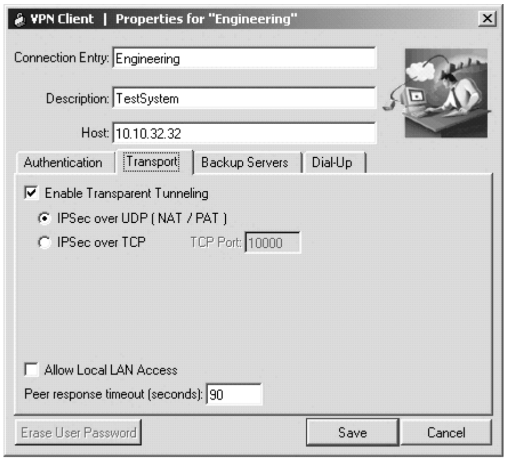
Step 16 Choose the name of the certificate that you are using from the menu.

If the field reads “No Certificates Installed” and is shaded, you must enroll for a certificate before you can use this feature.

Configure Transparent Tunneling

This section describes how to enable transparent tunneling.

Configure Transparent Tunneling



VPN Client | Properties for "Engineering"

Connection Entry: Engineering

Description: TestSystem

Host: 10.10.32.32

☑ Enable Transparent Tunneling

● IPSec over UDP (NAT / PAT)

○ IPSec over TCP   TCP Port: 10000

☐ Allow Local LAN Access

Peer response timeout (seconds): 90

SNRS v2.0—4-53

Next, configure transparent tunneling by completing the fields on the Transport tab.

Enabling Transparent Tunneling

Transparent tunneling allows secure transmission between the Cisco VPN Client and a secure gateway through a router serving as a firewall, which may also be performing NAT or PAT. Transparent tunneling encapsulates IP protocol 50 (ESP) traffic within User Datagram Protocol (UDP) packets and can allow for both ISAKMP (UDP 500) and IP protocol 50 to be encapsulated in TCP packets before they are sent through the NAT or PAT devices or firewalls. The most common application for transparent tunneling is behind a home router performing PAT.

The Cisco VPN Client also sends keepalives frequently, ensuring that the mappings on the devices are kept active.

Not all devices support multiple simultaneous connections behind them. Some cannot map additional sessions to unique source ports. Be sure to check with the vendor of your device to verify whether this limitation exists. Some vendors support IP protocol 50 PAT (IPsec pass-through), which might let you operate without enabling transparent tunneling.

To use transparent tunneling, the central-site group in the Cisco VPN device must be configured to support transparent tunneling

Transparent tunneling is enabled by default. To disable this parameter, uncheck the check box. It is recommended that you always keep this parameter checked.

Then choose a mode of transparent tunneling, over UDP or over TCP. The mode that you use must match the mode used by the secure gateway to which you are connecting. Either mode operates properly through a PAT device. Multiple simultaneous connections might work better with TCP, and if you are in an extranet environment, in general, TCP mode is preferable. UDP does not operate with stateful firewalls; therefore, in this case, you should use TCP.

Using IPsec over UDP (NAT or PAT)

To enable IPsec over UDP (NAT or PAT), click the **IPsec over UDP (NAT/PAT)** radio button. With UDP, the port number is negotiated. UDP is the default mode.

Using IPsec over TCP (NAT, PAT, or Firewall)

To enable IPsec over TCP, click the **IPsec over TCP** radio button. When using TCP, you must also enter the port number for TCP in the TCP port field. This port number must match the port number configured on the secure gateway. The default port number is 10000.

Allowing Local LAN Access

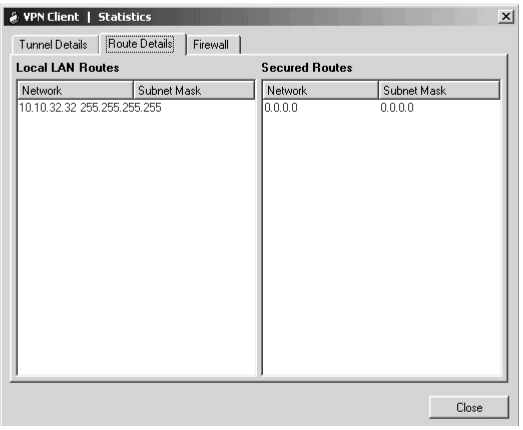
In a multiple network interface card (NIC) configuration, local LAN access pertains only to network traffic on the interface on which the tunnel was established. The Allow Local LAN Access parameter gives you access to the resources on your local LAN (printer, fax, shared files, or other systems) when you are connected through a secure gateway to a central-site VPN device. When this parameter is enabled and your central site is configured to permit it, you can access local resources while connected. When this parameter is disabled, all traffic from your client system goes through the IPsec connection to the secure gateway.

To enable this feature, check the **Allow Local LAN Access** check box; to disable it, uncheck the check box. If the local LAN that you are using is not secure, you should disable this feature. For example, you would disable this feature when you are using a local LAN in a hotel or airport.

A network administrator at the central site configures a list of networks at the client side that you can access. You can access up to 10 networks when this feature is enabled. When the Allow Local LAN Access feature is enabled and you are connected to a central site, all traffic from your system goes through the IPsec tunnel except traffic to the networks excluded from doing so (in the network list).

When this feature is enabled and configured on the Cisco VPN Client and permitted on the central-site VPN device, you can see a list of the local LANs available by looking at the routes table.

Routes Table



Local LAN Routes

Network	Subnet Mask
10.10.32.32	255.255.255.255

Secured Routes

Network	Subnet Mask
0.0.0.0	0.0.0.0

To display the routes table, complete these steps:

Step 1 Choose the **Status** menu, and choose **Statistics.**

Step 2 Choose **Route Details** from the Statistics dialog box.

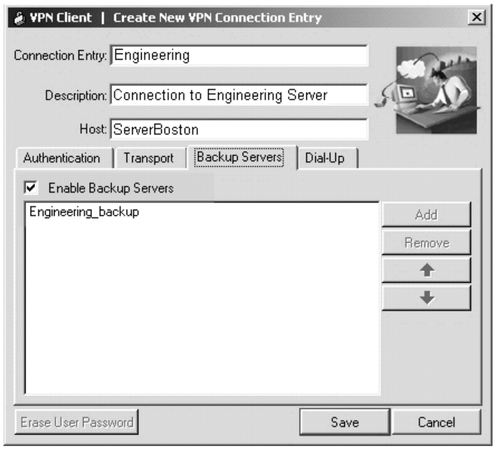
The routes table shows local LAN routes, which do not traverse the IPsec tunnel, and secured routes, which do traverse the IPsec tunnel to a central-site device. The routes in the local LAN routes column are for locally available resources.

Note This feature works on only one NIC—the same NIC as the tunnel.

Enable and Add Backup Servers

This section describes how to enable and add backup servers.

Enable and Add Backup Servers



Connection Entry: Engineering

Description: Connection to Engineering Server

Host: ServerBoston

☑ Enable Backup Servers

Engineering_backup

SNRS v2.0—4-55

The private network may include one or more backup VPN servers to use if the primary server is not available. Your system administrator tells you whether to enable backup servers. Information on backup servers can download automatically from the Cisco VPN concentrator, or you can manually enter this information.

To enable backup servers from the Cisco VPN Client, complete the following steps:

Step 1 Click the **Backup Servers** tab.

Step 3 Check the **Enable Backup Servers** check box. This box is not checked by default.

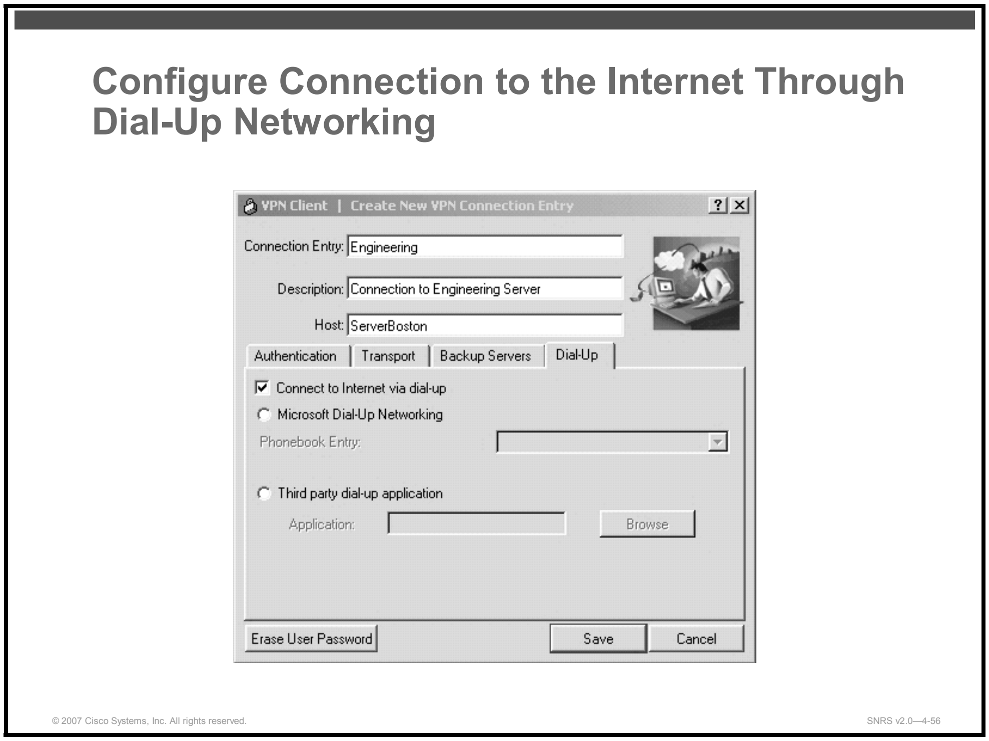
Step 4 Click **Add** to enter the backup server address.

Step 5 Enter the hostname or IP address of the backup server, using a maximum of 255 characters.

Step 6 To add more backup devices, repeat Steps 2, 3, and 4.

Configure Connection to the Internet Through Dialup Networking

This section describes how to configure the client to use a dial-up connection.



Configure Connection to the Internet Through Dial-Up Networking

VPN Client | Create New VPN Connection Entry

Connection Entry: Engineering

Description: Connection to Engineering Server

Host: ServerBoston

☑ Connect to Internet via dial-up

○ Microsoft Dial-Up Networking

Phonebook Entry:

○ Third party dial-up application

Application:

SNRS v2.0—4-56

To connect to a private network using a dialup connection, complete these steps:

Step 1 Use a dialup connection to your ISP to connect to the Internet.

Step 2 Use the Cisco VPN Client to connect to the private network through the Internet.

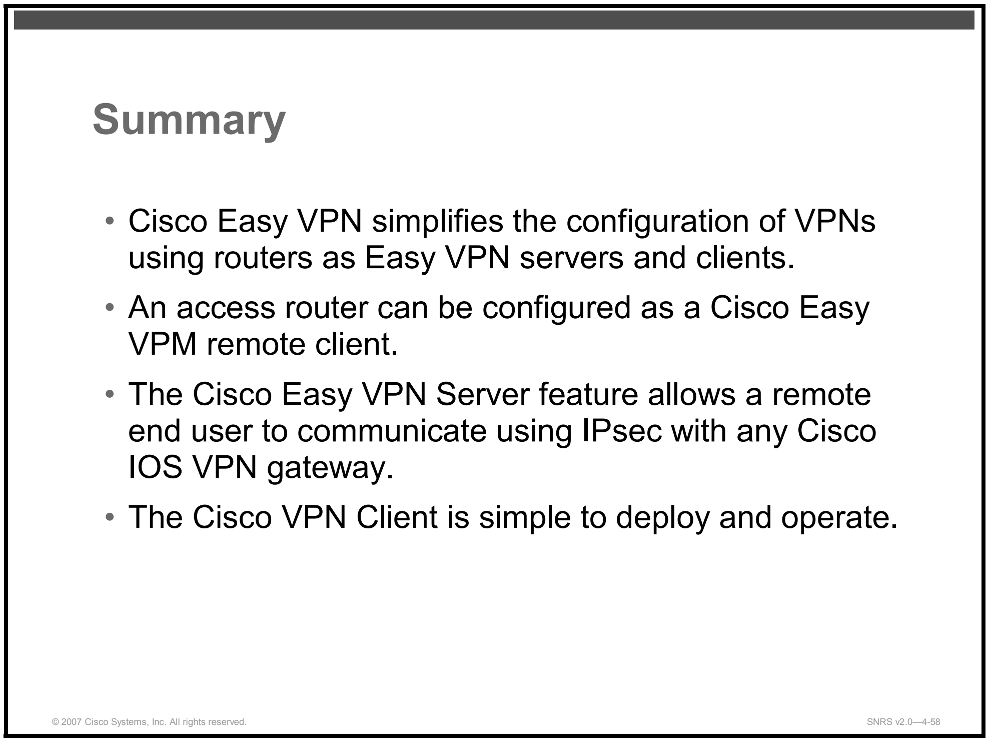
To enable and configure this feature, check the **Connect to the Internet via Dial-Up** check box. This box is not checked by default.

You can connect to the Internet using the Cisco VPN Client application in either of the following ways:

- Microsoft Dial-Up Networking (DUN)
- Third-party dialup program

Summary

This topic summarizes the key points that were discussed in this lesson.



Summary

- Cisco Easy VPN simplifies the configuration of VPNs using routers as Easy VPN servers and clients.
- An access router can be configured as a Cisco Easy VPM remote client.
- The Cisco Easy VPN Server feature allows a remote end user to communicate using IPsec with any Cisco IOS VPN gateway.
- The Cisco VPN Client is simple to deploy and operate.

SNRS v2.0—4-58

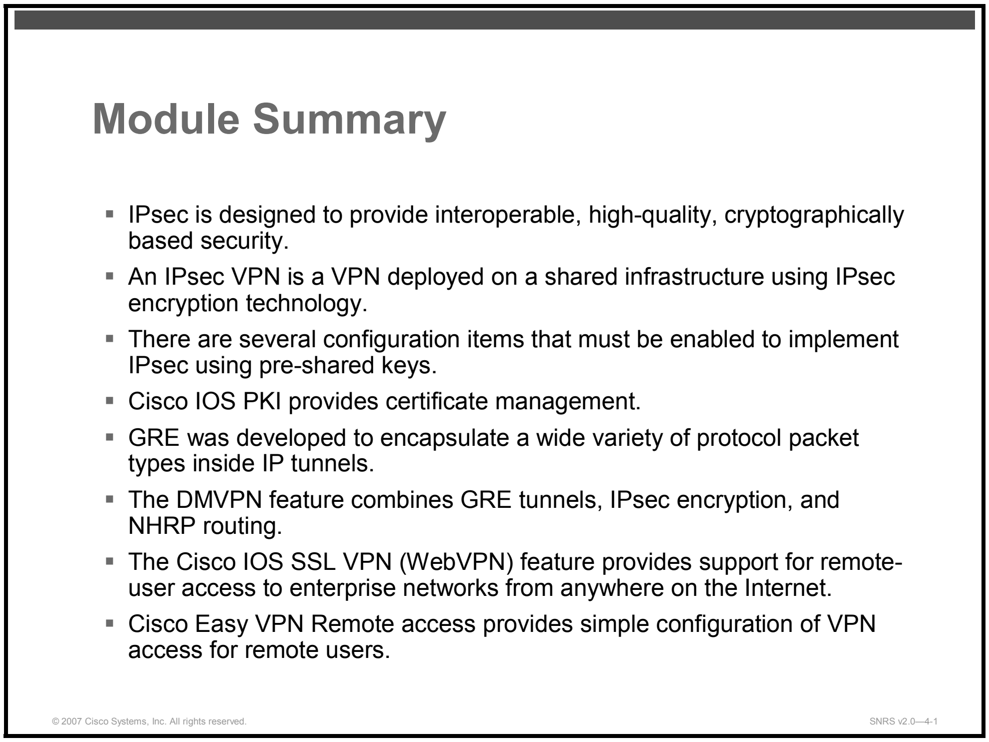
References

For additional information, refer to this resource:

- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4:* http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_book09186a008043360a.html
- *Configuring Cisco IOS Software Easy VPN IPsec Functionality:* http://www.cisco.com/en/US/products/ps6635/products_white_paper09186a00802341eb.shtml

Module Summary

This topic summarizes the key points that were discussed in this module.



Module Summary

- IPsec is designed to provide interoperable, high-quality, cryptographically based security.
- An IPsec VPN is a VPN deployed on a shared infrastructure using IPsec encryption technology.
- There are several configuration items that must be enabled to implement IPsec using pre-shared keys.
- Cisco IOS PKI provides certificate management.
- GRE was developed to encapsulate a wide variety of protocol packet types inside IP tunnels.
- The DMVPN feature combines GRE tunnels, IPsec encryption, and NHRP routing.
- The Cisco IOS SSL VPN (WebVPN) feature provides support for remote-user access to enterprise networks from anywhere on the Internet.
- Cisco Easy VPN Remote access provides simple configuration of VPN access for remote users.

There are several varieties of virtual private network (VPN) topologies available for secure remote access. In this module, you were introduced to IPsec and some of the protocols used in the standardized framework known as IPsec. You then examined Cisco's VPN access solutions such as:

- Site-to-site IPsec VPNs
  - Using pre-shared keys
  - Using PKI
- DMVPN
- GRE
- SSL VPN

References

For additional information, refer to this resource:

- Cisco Systems, Inc. *Cisco IOS Security Configuration Guide, Release 12.4: Dynamic Multipoint VPN* (DMVPN): http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_chapter09186a0080455c71.html.