CHAPTER 4

IPSec Authentication and Authorization Models

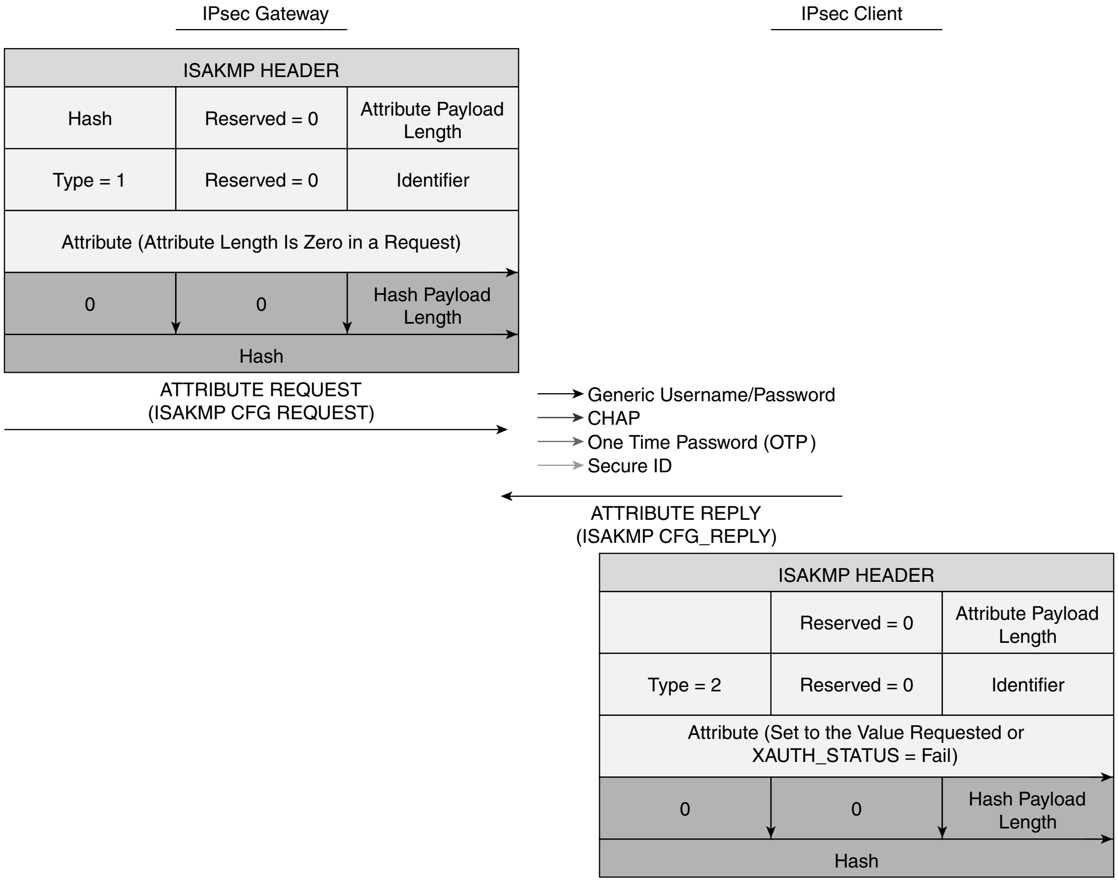
Telecommuting is increasingly becoming a part of everyday life, and IPSec connectivity is arguably the most popular connection method that telecommuters use to connect to a VPN across the public Internet. Given that telecommuters can be anywhere in the Internet, the successful completion of authorization and authentication is critical for granting access to the VPN. This chapter explores authentication and authorization models for the IPSec telecommuter.

Extended Authentication (XAUTH) and Mode Configuration (MODE-CFG)

Authentication schemes such as Remote Authentication Dial-In User Service (RADIUS) and SecureID are commonly used for providing secure remote access. It is highly desirable to leverage these authentication mechanisms for IPSec remote access. But Internet Key Exchange (IKE) protocol, which you learned about in Chapter 2, "IPSec Overview," does not provide a method to leverage these unidirectional authentication schemes. Extended Authentication, commonly referred to as XAUTH, was developed to leverage these legacy authentication schemes with IKE.

XAUTH provides an additional level of authentication by allowing the IPSec gateway to request extended authentication from remote users, thus forcing remote users to respond with their credentials before being allowed access to the VPN. It should be noted that XAUTH functions by first forming an IKE phase 1 SA using conventional IKE, and then by extending the IKE exchange to include additional user authentication exchanges. Figure 4-1 shows an XAUTH exchange using a generic username and password authentication scheme.

Figure 4-1 *Extended Authentication (XAUTH) Exchange*



As shown in Figure 4-1, XAUTH uses a Request/Reply mechanism to provide the extended authentication. The XAUTH process is terminated, either when the gateway starts a SET/ACK exchange, which includes an XAUTH_STATUS attribute, or when the remote device sends a XAUTH_STATUS attribute in a REPLY message.

The XAUTH protocol defines four message types that are exchanged between the remote user and the IPSec gateway. These messages carry various attributes for the extended authentication process to work. The four XAUTH message types are:

- ISAKMP_CFG_REQUEST
- ISAKMP_CFG_REPLY
- ISAKMP_CFG_SET
- ISAKMP_CFG_ACK

A description of the XAUTH message types follows:

- **ISAKMP_CFG_REQUEST**—This message is sent from the IPSec gateway to the IPSec client requesting extended authentication of the client.
- **ISAKMP_CFG_REPLY**—This message must contain the filled-in authentication attributes that were requested by the gateway or, if the proper authentication attributes cannot be retrieved, this message must contain the XAUTH_STATUS attribute with a value of FAIL.
- **ISAKMP_CFG_SET**—This message is sent from the gateway and is used only to state the success or failure of the authentication.
- **ISAKMP_CFG_ACK**—This message is sent from the IPSec client, acknowledging receipt of the authentication result.

The XAUTH message types defined above carry various attributes. A brief description of the attributes is shown in Table 4-1.

Table 4-1 *XAUTH Attributes*

Attribute	Description
XAUTH_TYPE	This attribute describes the type of extended authentication method requested. Four authentication methods are defined in the protocol: Generic, RADIUS_CHAP, One Time password (OTP), and Secure ID. This is an optional attribute for the ISAKMP_CFG_REQUEST and ISAKMP_CFG_REPLY messages. The XAUTH_TYPE in a REPLY must be identical to the XAUTH_TYPE in the REQUEST. However, an XAUTH transaction may have multiple REQUEST/REPLY pairs with different XAUTH_TYPE values in each pair.
XAUTH_USER_NAME	The username may be any unique identifier of the user, such as a login name, an email address, or a X.500 Distinguished Name.
XAUTH_USER_PASSWORD	The user's password.
XAUTH_PASSCODE	A token card's passcode.

continues

Table 4-1 *XAUTH Attributes (Continued)*

Attribute	Description
XAUTH_MESSAGE	A textual message from the gateway to the IPSec client. The message may contain a textual challenge or instruction.
XAUTH_CHALLENGE	A challenge string sent from the gateway to the IPSec client to be included in its calculation of a password. This attribute should be sent only in an ISAKMP_CFG_REQUEST message. Typically, the XAUTH_TYPE attribute dictates how the receiving device should handle the challenge. For example, RADIUS-CHAP uses the challenge to hide the password.
XAUTH_STATUS	A variable that is used to denote authentication success (OK=1) or failure (FAIL=0). This attribute *must* be sent in the ISAKMP_CFG_SET message, in which case it may be set to either OK or FAIL, and *may* be sent in a REPLY message by a remote peer, in which case it *must* be set to FAIL.

When a remote access user connects to an IPSec gateway and XAUTH is required by the gateway, configuration on the gateway initiates the XAUTH messages before IKE phase 2 negotiation begins. If the remote access client does not have support for the authentication method requested by the gateway, the client would send back a REPLY with the XAUTH_STATUS attribute set to FAIL, thus failing the authentication.

Example 4-1 shows the configuration of XAUTH using the RADIUS/AAA authentication method.

Example 4-1 *Cisco IOS XAUTH Configuration on the IPSec Gateway*

```
vpn-gw1-east#
!
hostname vpn-gw1-east
!
username ezvpn password 0 east
username ezvpn1@vpngroup password 0 ezvpn1east
username ezvpn2@vpngroup password 0 ezvpn2east
aaa new-model
!
aaa authentication login vpn local
aaa authorization network vpn local
aaa session-id common
ip subnet-zero
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp keepalive 10 10
```

Example 4-1 *Cisco IOS XAUTH Configuration on the IPSec Gateway (Continued)*

```
!
crypto ipsec transform-set vpn esp-3des esp-sha-hmac
!
crypto dynamic-map dynamic 1
 set transform-set vpn
 reverse-route remote-peer 9.1.1.33
!
!
crypto map vpn client authentication list vpn
crypto map vpn isakmp authorization list vpn
crypto map vpn client configuration address respond
crypto map vpn 3 ipsec-isakmp dynamic dynamic
```

The addition of the following command on the crypto map enables XAUTH and triggers the XAUTH transaction after IKE phase 1 and before IKE phase 2:

```
crypto map map-name client authentication list list-name
```

As you learned in Chapter 2, "IPSec Overview," a very common deployment scenario for IPSec telecommuters is the use of IKE pre-shared key authentication with Aggressive Mode. The primary motivation for this scenario is that the IP address of an IPSec remote access user connecting to an IPSec gateway over the public Internet is typically not known in advance to the gateway. In most deployments using pre-shared keys, a single shared group key is used for all users of the VPN. What this means is that without employing some form of additional user authentication, there is no way to verify that the person connecting with that VPN client is indeed a valid user.

Imagine, for example, a situation where a laptop with a VPN client is stolen—because the VPN client is already configured with a valid group key, anyone with the laptop can connect to the VPN without any problems, as no further authentication is required! Extended Authentication (XAUTH) is widely employed to address this serious security gap. XAUTH forces users to identify themselves with a user id and a password *after* the group pre-shared key has been verified.

XAUTH is also referred to as "two factor authentication." The password could be a "one-time password" (for example, from a SecureID card) adding further security to such a deployment. Although the usage of XAUTH is very common and desired for the telecommuter scenario using pre-shared keys and Aggressive Mode, it can also be used with Main Mode and other authentication methods such as digital certificates. It is important to note that although XAUTH is deployed very commonly, it has not been established as a standard by the IPSec working group in the IETF, which means that it may present interoperability issues among different vendor implementations.

Mode-Configuration (MODECFG)

In remote access scenarios, it is highly desirable to be able to push configuration information such as the private IP address, a DNS server's IP address, and so forth, to the client. The IPSec Mode-configuration (MODECFG) allows this functionality. Configuration for MODECFG using Cisco IOS is shown in Example 4-2.

Example 4-2 *Cisco IOS MODECFG Configuration on the IPSec Gateway*

```
vpn-gw1-east#
!
hostname vpn-gw1-east
!
username ezvpn password 0 east
username ezvpn1@vpngroup password 0 ezvpn1east
username ezvpn2@vpngroup password 0 ezvpn2east
aaa new-model
!
aaa authentication login vpn local
aaa authorization network vpn local
aaa session-id common
ip subnet-zero
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp keepalive 10 10
!
crypto isakmp client configuration group vpngroup
 key ciscoezvpn
 dns 10.1.1.10
 wins 10.1.1.11
 pool vpnpool
 include-local-lan
 backup-gateway 9.1.1.36
!
crypto ipsec transform-set vpn esp-3des esp-sha-hmac
!
crypto dynamic-map dynamic 1
 set transform-set vpn
 reverse-route remote-peer 9.1.1.33
!
!
crypto map vpn client authentication list vpn
crypto map vpn isakmp authorization list vpn
crypto map vpn client configuration address respond
crypto map vpn 3 ipsec-isakmp dynamic dynamic
```

Some of the key attributes that can be pushed to a remote user using MODECFG follow:

- **INTERNAL_IP4_ADDRESS**, **INTERNAL_IP6_ADDRESS**—Specifies an address within the internal network. The requested address is valid until the expiration of the ISAKMP SA that was used to secure the request. The address may also expire when the IPSec phase 2 SA expires, if the request is associated with a phase 2 negotiation.
- **INTERNAL_IP4_NETMASK**, **INTERNAL_IP6_NETMASK**—The internal network's netmask.
- **INTERNAL_IP4_DNS**, **INTERNAL_IP6_DNS**—Specifies an address of a DNS server or multiple DNS servers within the network. The responder may respond with zero, one, or more DNS server attributes.
- **INTERNAL_IP4_NBNS**, **INTERNAL_IP6_NBNS**—Specifies an address of a NetBios Name Server (NBNS) within the network. Multiple NBNSs may be requested. The responder may respond with zero, one, or more NBNS attributes.

Like XAUTH, MODECFG is not a standard of the IPSec working group in the IETF. Although Cisco defined this protocol and most client implementations work with the Cisco implementation, given that this not a standard, there are no guarantees for interoperability.

Easy VPN (EzVPN)

As you saw in Chapter 2, "IPSec Overview," for an IPSec tunnel to be established between two peers, there is a significant amount of configuration required on both peers. This includes IPSec policies, Diffie-Hellman parameters, encryption algorithms, and so on. In a large corporate environment with hundreds of sites, managing the IPSec configuration can get quite tedious. The Cisco Easy VPN feature, also known as EzVPN, eases IPSec configuration by allowing an almost no-touch configuration of the IPSec client.

EzVPN uses the Unity client protocol, which allows most IPSec VPN parameters to be defined at an IPSec gateway, which is also the EzVPN server. When an EzVPN client initiates an IPSec tunnel connection, the EzVPN server pushes the IPSec policies and other attributes required to form the IPSec tunnel to the EzVPN client and creates the corresponding IPSec tunnel connection. The tunnel on the EzVPN client can be initiated automatically or manually, or it could be traffic triggered, depending on the configuration or type of EzVPN client used. Minimal configuration is required at the EzVPN client. EzVPN provides the following general functions in order to simplify the configuration process:

- **Negotiating tunnel parameters**—This is done with encryption algorithms, SA lifetimes, and so on.
- **User authentication**—This entails validating user credentials by way of XAUTH.
- **Automatic configuration**—Performed by pushing attributes such as IP address, DNS, WINs, and so on, using MODECFG.

NOTE The term EzVPN client is used for both Cisco Unity VPN clients, called EzVPN software clients, and the Unity client protocol running on smaller Cisco routers like the 800, 1700, and 2600 series, commonly referred to as EzVPN hardware clients.

The Cisco Easy VPN feature supports two modes of operation:

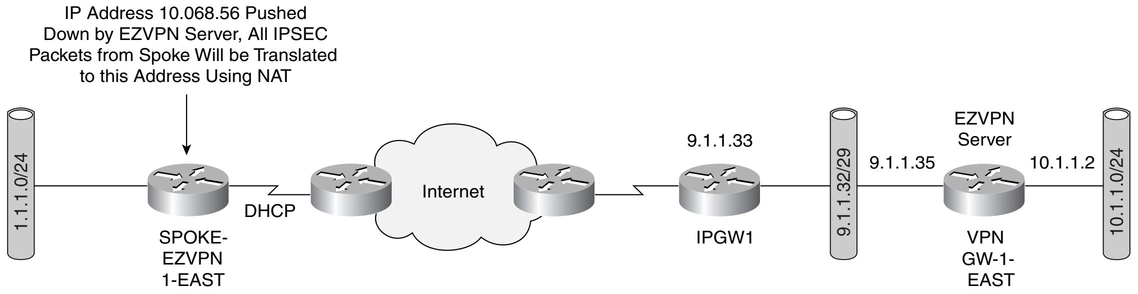
- Client Mode
- Network Extension Mode

You will examine each of these modes in detail in the following sections.

EzVPN Client Mode

Figure 4-2 shows an IPSec Unity client configured for Client Mode in order to establish an IPSec VPN tunnel to the gateway.

Figure 4-2 *EzVPN IPSec Client Mode Connection*


IP Address 10.068.56 Pushed
Down by EZVPN Server, All IPSEC
Packets from Spoke Will be Translated
to this Address Using NAT
1.1.1.0/24
SPOKE-
EZVPN
1-EAST
DHCP
Internet
9.1.1.33
IPGW1
9.1.1.32/29
9.1.1.35
EZVPN
Server
10.1.1.2
VPN
GW-1-
EAST
10.1.1.0/24

EzVPN Client Mode is also known as Network/Port Address Translation (NAT/PAT) Mode. In this mode, all traffic from the client side uses a single IP address for all hosts on the private network. In Figure 4-2, all traffic from the hosts on the FastEthernet interface on the EzVPN client is translated by NAT to a source IP address of 10.0.68.5, which is assigned by the EzVPN server as an attribute using MODECFG. The client keeps track of the mappings so that it can be forwarded to the correct host on the private network.

The configuration of the EzVPN hardware client is shown in Example 4-3.

Example 4-3 *EzVPN Client Mode Configuration*

```
spoke-ezvpn1-east#
!
hostname spoke-ezvpn1-east
!
```

Example 4-3 *EzVPN Client Mode Configuration (Continued)*

```
crypto ipsec client ezvpn vpn
 connect auto
 group vpngroup key ciscoezvpn
 local-address Ethernet0
 mode client
 peer 9.1.1.35
 username ezvpn1@vpngroup password  ezvpn1east
!
!
interface Ethernet0
 ip address dhcp
 load-interval 30
 half-duplex
 crypto ipsec client ezvpn vpn outside
!
interface FastEthernet0
 ip address 1.1.1.1 255.255.255.0
 load-interval 30
 speed 100
 full-duplex
 no keepalive
 crypto ipsec client ezvpn vpn inside
!
ip route 0.0.0.0 0.0.0.0 Ethernet0
!
end
```

Notice that in the EzVPN client configuration, none of the IPSec policies, encryption algorithms, and so forth are configured. Example 4-4 shows how to monitor an EzVPN client configuration.

Example 4-4 *Verification of EzVPN Client Mode Configuration*

```
spoke-ezvpn1-east#show crypto isakmp sa
dst             src             state          conn-id slot
9.1.1.35        9.1.1.146       QM_IDLE              4    0
spoke-ezvpn1-east#show crypto ipsec client ezvpn
Easy VPN Remote Phase: 2
Tunnel name : vpn
Inside interface list: FastEthernet0,
Outside interface: Ethernet0
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 10.0.68.56
Mask: 255.255.255.255
DNS Primary: 10.1.1.10
NBMS/WINS Primary: 10.1.1.11
```

The configuration of the EzVPN server is shown in Example 4-5.

Example 4-5 *EzVPN Server-side Configuration*

```
vpn-gw1-east#
!
hostname vpn-gw1-east
!
username ezvpn password 0 east
username ezvpn1@vpngroup password 0 ezvpn1east
username ezvpn2@vpngroup password 0 ezvpn2east
aaa new-model
!
aaa authentication login vpn local
aaa authorization network vpn local
aaa session-id common
ip subnet-zero
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp keepalive 10 10
!
crypto isakmp client configuration group vpngroup
 key ciscoezvpn
 dns 10.1.1.10
 wins 10.1.1.11
 pool vpnpool
 include-local-lan
 backup-gateway 9.1.1.36
!
!
crypto ipsec transform-set vpn esp-3des esp-sha-hmac
!
crypto dynamic-map dynamic 1
 set transform-set vpn
 reverse-route remote-peer 9.1.1.33
!
!
crypto map vpn client authentication list vpn
crypto map vpn isakmp authorization list vpn
crypto map vpn client configuration address respond
crypto map vpn 3 ipsec-isakmp dynamic dynamic
!
!
interface Loopback0
 ip address 9.2.1.100 255.255.255.255
!
interface FastEthernet0/0
 ip address 9.1.1.35 255.255.255.248
 duplex full
 crypto map vpn
!
```

Example 4-5 *EzVPN Server-side Configuration (Continued)*

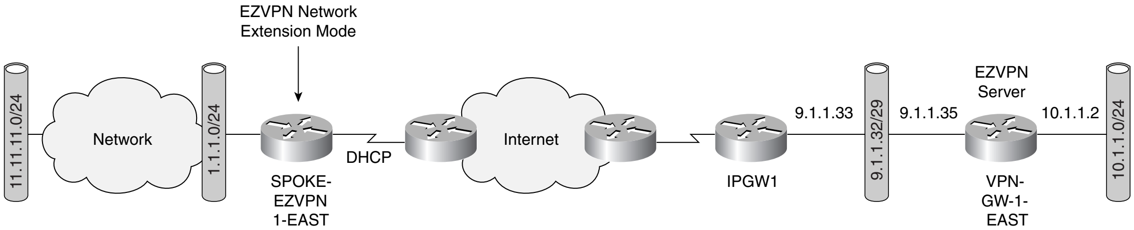
```
interface FastEthernet2/0
 ip address 100.1.1.147 255.255.255.0
 duplex full
!
interface FastEthernet4/0
 ip address 10.1.1.1 255.255.255.0
 duplex full
!
router ospf 1
 log-adjacency-changes
 redistribute static subnets
 network 10.1.1.0 0.0.0.255 area 0
!
ip local pool vpnpool 10.0.68.1 10.0.68.100
ip classless
ip route 0.0.0.0 0.0.0.0 9.1.1.33
!
radius-server host 100.1.1.4 auth-port 1645 acct-port 1646
radius-server key cisco
end
```

The IOS command **crypto isakmp client configuration group vpngroup** defines the attributes for the VPN group that was assigned to the EzVPN client.

Network Extension Mode

Figure 4-3 shows an EzVPN client in Network Extension Mode. This mode allows the EzVPN client to present a full, routable network to the tunneled network.

Figure 4-3 *EzVPN IPSec Network Extension Mode Connection*



IPSec encapsulates all traffic from the EzVPN client's private network, which is marked as 'inside' to networks behind the IPSec gateway. Therefore, devices behind the gateway have direct access to devices on the EzVPN client's private network via the tunnel and vice versa without the need for NAT or PAT. As there is no reason for NAT or PAT, the EzVPN server does not push down an IP address for tunneled traffic, but all other attributes like ACLs, DNS, and

WINS, can be pushed down. Example 4-6 shows the configuration of Network Extension Mode on the EzVPN client shown in Figure 4-3.

Example 4-6 *EzVPN Client Configuration for Network Extension Mode*

```
spoke-ezvpn1-east#
!
hostname spoke-ezvpn1-east
!
crypto ipsec client ezvpn vpn
 connect auto
 group vpngroup key ciscoezvpn
 local-address Ethernet0
 mode network-extension
 acl 100
 peer 9.1.1.35
 username ezvpn1@vpngroup password  ezvpn1east
!
interface Ethernet0
 ip address dhcp
 load-interval 30
 half-duplex
 crypto ipsec client ezvpn vpn outside
!
interface FastEthernet0
 ip address 1.1.1.1 255.255.255.0
 load-interval 30
 speed 100
 full-duplex
 no keepalive
 crypto ipsec client ezvpn vpn inside
!
access-list 100 permit 11.1.1.0 0.0.0.255 any
ip route 0.0.0.0 0.0.0.0 dhcp
```

Note the ACL 100 under the mode network-extension in the configuration. The ACL 100 permits networks behind the “inside” network (11.1.1.0/24) and allows traffic to and from these subnets to be encrypted. Without the access list, only traffic to and from the “inside” subnet (1.1.1.0/24) is encrypted.

NOTE The EzVPN server configuration is the same for both Client Extension Mode and Network Extension Mode. The client configuration determines which mode is being used.

Redundancy is always an integral part of any IPSec design and, in the case of EzVPN dead peer detection along with backup peer list, makes such a design possible. Example 4-7 shows the client configuration with multiple EzVPN server peer addresses manually configured on the client. An alternate mechanism to provide EzVPN server redundancy is to push the backup server's address list down to the client as an attribute. Dead peer detection is on by default on the EzVPN clients.

Example 4-7 *EzVPN Server Redundancy*

```
spoke-ezvpn1-east#
!
hostname spoke-ezvpn1-east
!
crypto ipsec client ezvpn vpn
 connect auto
 group vpngroup key ciscoezvpn
 local-address Ethernet0
 mode network-extension
 peer 9.1.1.35
 peer 9.1.1.45
 username ezvpn1@vpngroup password  ezvpn1east
```

EzVPN can be combined with XAUTH and MODECFG to provide extended authentication and thereby push all MODECFG attributes to the client. The attributes to be pushed can be defined locally on the EzVPN server or defined on a AAA server and defined either on a per-group or per-user basis. Example 4-8 shows the policies configured on a AAA server on a group basis.

Example 4-8 *Per-Group–based Policy Configuration on a AAA Server*

```
vpngroup Password = "cisco", Service-Type = Outbound
 cisco-avpair = "ipsec:tunnel-type=ESP"
 cisco-avpair = "ipsec:key-exchange=ike"
 cisco-avpair = "ipsec:tunnel-password=ciscoezvpn"
 cisco-avpair = "ipsec:addr-pool=vpnpool"
 cisco-avpair = "ipsec:default-domain=cisco"
 cisco-avpair = "ipsec:inacl=101"
 cisco-avpair = "ipsec:access-restrict=fastethernet 0/0"
 cisco-avpair = "ipsec:group-lock=1"
cisco-avpair = "ipsec:backup-server=9.1.1.35"

 cisco-avpair = "ipsec:dns-servers=10.1.1.10"
 cisco-avpair = "ipsec:firewall=1"
 cisco-avpair = "ipsec:include-local-lan=1"
 cisco-avpair = "ipsec:save-password=1"
 cisco-avpair = "ipsec:wins-servers=10.1.1.11"
 cisco-avpair = "ipsec:max-users = 100"
 cisco-avpair = "ipsec:max-logins = 2"
```

NOTE When AAA is used for pushing the attributes, AAA between EzVPN server and the AAA server requires a hard-coded password of **cisco**. Therefore, notice the use of the string **cisco** as the vpngroup password. This is a requirement of the Cisco AAA implementation.

Example 4-9 shows the configuration of the policy attributes locally on the EzVPN server on a per-group basis.

Example 4-9 *Per-Group EzVPN Policy Attributes Configured Locally on an EzVPN Server*

```
crypto isakmp client configuration group vpngroup
 key ciscoezvpn
 dns 10.1.1.10
 wins 10.1.1.11
 domain cisco.com
 pool vpnpool
 group-lock
 save-password
 include-local-lan
 pfs
 backup-gateway 9.1.1.36
 max-users 100
 max-logins 2
 access-restrict fastEthernet 0/0
```

You've already seen described some attributes such as IP address, WINS, DNS, and backup server in the previous section. For a complete list of attributes and their description, refer to Cisco IOS EzVPN documentation. A couple of very useful attributes for telecommuter scenarios are max-logins and include-local-lan. The max-logins attribute allows the administrator to restrict the number of simultaneous IPSec connections from the same user to the gateway. The include-local-lan attribute is very useful in scenarios in which the telecommuter's LAN has resources, such as printers, attached to the LAN and access to these resources is required when the VPN tunnel is up.

The attributes may also be applied on a per-user basis. A user attribute overrides a group attribute value. These attributes are retrieved at the time user authentication occurs using XAUTH, and are then combined with group attributes and applied during Mode-Configuration. User-based attributes are available only if RADIUS is used as the database. Example 4-10 shows the RADIUS attributes that must be configured on a per-user basis.

Example 4-10 *User-Based Policy Control Using AAA*

```
ezvpn1@vpngroup Password = "ezvpn1east"
framed-Ip-Address=10.0.68.1
ipsec:user-save-password=1
ipsec:user-include-local-lan=1
ipsec:user-vpn-group=cisco
```

Note that EzVPN using the Cisco Unity protocol is not an IETF standard. The Cisco proprietary Unity protocol is supported by Cisco IOS devices, VPN 3000, and PIX devices for interoperability across Cisco devices.

Digital Certificates for IPSec VPNs

You have learned from Chapter 2, "IPSec Overview," that pre-shared keys and digital certificates are two primary authentication methods in IKE that can be used in the context of IPSec VPN deployments. You also saw that pre-shared keys with IKE Main Mode cannot be used for remote access scenarios in which the responder does not know the source IP address of the initiator in advance. Although Aggressive Mode can alleviate this issue, given that Aggressive Mode does not offer any form of ID protection, such a solution may not be acceptable. You also saw XAUTH provide an additional level of authentication for Aggressive Mode with pre-shared keys. Another solution for such a scenario is the use of digital certificates instead of pre-shared keys.

In addition to solving the authentication issue for remote access users, digital certificate–based authentication is also becoming increasingly popular for large IPSec VPN site-to-site deployments because it is more scalable than pre-shared keys. Without digital signatures, users must either manually exchange public keys or secrets between each pair of devices that use IPSec to protect communications. This is cumbersome because when a new device is added to the network, users are required to make configuration changes on every other device it securely communicates with. By using digital certificates, users simply enroll each new device with a Certificate Authority, and none of the other devices need modification. When the new device attempts an IPSec connection, IKE automatically exchanges certificates with the peer and the devices authenticate each other, thus making a large-scale IPSec VPN deployment very scalable using digital certificates. An entire book can be written on digital certificates and PKI. You will be introduced to some of the components of PKI in the remainder of this chapter.

Digital Certificates

Digital certificates provide a means to digitally authenticate devices and individual users. These certificates act kind of like an online passport—they are tamper proof and cannot be forged. An individual that wishes to send encrypted data obtains a digital certificate from a *Certificate Authority (CA)*. The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available. The recipient of the encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA, and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply. Public key infrastructure (PKI) is the enabler for managing digital certificates for IPSec VPN deployment. The most widely used format for digital certificates is X.509, which is supported by Cisco IOS.

Certificate Authority—Enrollment

The Certificate Authority is the entity that issues the digital certificate. Enrollment is the process of obtaining a new certificate from a certificate authority. The IOS command **crypto ca trustpoint** is used to declare the specific CA that the router should use for enrollment.

As PKI is deployed on larger networks, enrollment becomes a big problem if it requires manual enrollment. The Cisco auto-enroll feature will be useful for this situation. With this feature, an IOS router can be configured to periodically contact the CA and request a new certificate. Auto enrollment may be configured to generate new encryption keys or to continue to use existing keys. Routers can initially auto-enroll with CA and when the certificate lifetime expires, the router re-enrolls automatically. Example 4-11 shows the configuration snippet for enrolling a Cisco router to a CA as well as the auto-enrollment command.

Example 4-11 *Cisco IOS Configuration for Enrolling with CA*

```
crypto ca trustpoint Public_CA
 enrollment url http://100.1.1.5:80
subject-name OU=Engineering., O=ABC

 auto-enroll 95 regenerate
 password cisco
 rsakeypair public-ca
```

Auto enrollment makes sure that the router re-enrolls with the CA after the certificate lifetime expires. However, while re-enrollment is occurring, new incoming IKE connections cannot be established because the existing certificate and key pairs are deleted immediately after the new key is generated, and the new key does not have a certificate to match it until the enrollment is complete. A simple workaround is to have the router re-enroll to the CA *before* the certificate lifetime expires, using the IOS command **auto enroll <*percent*> regenerate**."

It's possible that a router may be required to enroll with multiple certificate servers. Each CA server has an independent policy and may have different requirements as to general versus special purpose certificates or key length. Also, although it does not substantially affect the security of the system, VPN providers would like to use different key pairs for each VPN. Using the same key pair gives the impression of lower security, which is detrimental to end-user satisfaction. Example 4-12 shows an example of enrollment to multiple CAs using multiple key pairs.

Example 4-12 *Enrollment with Multiple CAs Using Multiple Key Pairs*

```
vpn-gw1-east#show running-config
!
crypto ca trustpoint Public_CA
 enrollment url http://100.1.1.5:80
 revocation-check crl
 rsakeypair public-ca
 match certificate engg
!
crypto ca trustpoint IOS_CA
```

Example 4-12 *Enrollment with Multiple CAs Using Multiple Key Pairs (Continued)*

```
 enrollment url http://100.1.1.179:80
 revocation-check none
 rsakeypair ios-ca
!

vpn-gw1-east#show cry ca trustpoints
Trustpoint Public_CA:
    Subject Name:
    cn=Certificate Manager
    ou=nsite-rtp
    o=cisco-rtp
    l=rtp
    st=nc
    c=US
          Serial Number: 01
    Certificate configured.
    CEP URL: http://100.1.1.5

Trustpoint IOS_CA:
    Subject Name:
    cn=Certmanager O\=cisco OU\=nsite
          Serial Number: 01
    Certificate configured.
    CEP URL: http://100.1.1.179
```

Certificate Revocation

Situations may arise in which the certificate issued by a CA may need to be revoked before the lifetime of the certificate expires. One such situation would be if the certificate itself is known to have been compromised. Yet another more common situation in the context of IPSec VPNs is when temporary access to a VPN is desired. For example, assume that a user that belongs to Company A may want temporary access to Company B's VPN, and is granted a certificate for access. The access would need to be revoked after a short time in order to maintain integrity of the network.

Another precaution maintained prevents users whose certificate has been revoked from being granted access to a VPN. To achieve this, a list of certificates that have been revoked, known as a Certificate Revocation List (CRL), is sent periodically from the CA to the IPSec gateway. When an incoming IKE session is initiated for a user whose certificate is revoked, the CRL will be checked to see if the certificate is valid; if the certificate is revoked, IKE will fail and access to the VPN will be denied. Example 4-13 shows Cisco IOS configuration to check for a revoked certificate in the CRL.

Example 4-13 *Cisco IOS Configuration for Checking Certificate Status Using CRL*

```
crypto ca trustpoint Public_CA
 enrollment url http://100.1.1.5:80
 revocation-check crl
 rsakeypair public-ca
 match certificate eng
```

Once the CRL is obtained by the VPN endpoint from the CA, it will maintain it until the Certificate Revocation timer expires or if the endpoint reloads.

An alternate mechanism for checking the validity of a certificate is the use of the Online Certificate Status Protocol (OCSP). Unlike CRLs, which provide only periodic certificate status checks, OCSP can provide timely information regarding the status of a certificate. Example 4-14 shows the configuration to enable OCSP to check certificate status.

Example 4-14 *Cisco IOS Configuration for Checking Certificate Status Using OCSP*

```
crypto ca trustpoint Public_CA
 enrollment url http://100.1.1.5:80
 revocation-check ocsp
 rsakeypair public-ca
 match certificate eng
```

If the deployment needs more granular control over the permission to connect users (even those with valid certificates), certificate-based ACLs may be used. For example, there may be situations in which you wish to allow only a subset of users with a valid certificate to connect to the VPN. The configuration in Example 4-15 allows only users whose certificates contain “Engineering” in the subject name.

Example 4-15 *Cisco IOS Configuration for Certificate-Based ACLs*

```
crypto ca certificate map eng 10
 subject-name co Engineering

crypto ca trustpoint Public_CA
match certificate eng
 enrollment url http://100.1.1.5:80
 auto-enroll 95 regenerate
 password cisco
 rsakeypair public-ca
```

Digital certificates as an authentication method for large-scale IPSec VPNs is becoming increasingly popular for both remote access and site-to-site deployments. You learned in this section that the use of digital certificates requires some form of PKI infrastructure such as a CA server. Network administrators should be aware that managing the PKI infrastructure and digital certificates can be much more complicated than username-based password management. The benefits to be gained by the additional work by network administration should be carefully considered before using digital certificates.

Summary

This chapter covered IPSec features and mechanisms that are primarily targeted at the authentication of remote access users. You learned about XAUTH, which provides extended authentication for IPSec telecommuters by using authentication schemes such as RADIUS. MODECFG uses a push model to push attributes to the IPSec client. You also saw how EzVPN greatly simplifies configuration for IPSec clients and allows central configuration of the IPSec policies. You explored the use of digital certificates and PKI, which are becoming increasingly popular for scalable IPSec deployments for both remote access and site-to-site VPNs.