- IPSEC Site-to-Site VPNs on a PIX/ASA Firewall -

Configuring the ISAKMP Policy

The first step in creating an IPSEC Tunnel is to enable ISAKMP on a specific PIX/ASA interface:

Pixfirewall(config)# isakmp enable outside

Next, an ISAKMP policy must be created, which defines the algorithms and protocols applied to the ISAKMP SA during IKE Phase 1. To create the ISAKMP policy:

Pixfirewall(config)# isakmp policy 1 encryption des
Pixfirewall(config)# isakmp policy 1 hash sha
Pixfirewall(config)# isakmp policy 1 authentication rsa-sig
Pixfirewall(config)# isakmp policy 1 group 1
Pixfirewall(config)# isakmp policy 1 lifetime 86400

The isakmp policy command is used to define ISAKMP parameters. The 1 is a priority number, as we can have multiple ISAKMP policies. The lower the number, the higher priority the policy is.

The above values for each parameter are the default values. The following table details every possible option:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>des, 3des, aes, aes-192, aes-256</td>
</tr>
<tr>
<td>Hash</td>
<td>md5, sha</td>
</tr>
<tr>
<td>Authentication</td>
<td>pre-share, rsa-sig</td>
</tr>
<tr>
<td>Group</td>
<td>1, 2, 5</td>
</tr>
</tbody>
</table>

To view all configured ISAKMP policies:

Pixfirewall(config)# show isakmp policy

* * *
**Configuring Authentication (Pre-Shared Keys)**

If using a **pre-shared key** for authentication, a key string must be specified, pointing to the remote host’s IP address. To configure the pre-shared key on a PIX OS 6.0 device:

```
Pixfirewall(config)# isakmp policy 1 authentication pre-share
Pixfirewall(config)# isakmp key MYKEY address 77.1.1.1
```

To configure the pre-shared key on a PIX OS 7.0 device:

```
asafirewall(config)# tunnel-group 77.1.1.1 type ipsec-l2l
asafirewall(config)# tunnel-group 77.1.1.1 ipsec-attributes
asafirewall(config-tunnel-ipsec)# pre-shared-key MYKEY
```

Remember, both the shared key, and the ISAKMP policy must match on both peers for a session to be established. To view configured ISAKMP policies:

```
Pixfirewall(config)# show isakmp policy
```

**Configuring Authentication (Digital Signatures)**

Pre-shared keys are the **simplest** method of authentication. Much more configuration is required if we want to use **RSA Digital Signatures**. The hostname and domain-name for the PIX/ASA Firewall must be specified, as this information is included in the blank certificate sent to the CA:

```
Pixfirewall(config)# isakmp policy 1 authentication rsa-sig
Pixfirewall(config)# hostname Pixfirewall
Pixfirewall(config)# domain-name MYDOMAIN.COM
```

Next, the RSA key must be generated:

```
Pixfirewall(config)# ca generate rsa key 2048
```

Next, the Certificate Authority must be identified:

```
Pixfirewall(config)# ca identity MYCA 192.168.1.5
Pixfirewall(config)# ca configure MYCA ca 1 100
```

Next, the CA must be **authenticated**, to ensure its validity:

```
Pixfirewall(config)# ca authenticate MYCA
```
Configuring Authentication (Digital Signatures) (continued)

Next, a certificate must actually be requested. In other words, the certificate must be digitally signed.

    Pixfirewall(config)# ca enroll MYCA CAPASSWORD

Finally, the certificate must be saved:

    Pixfirewall(config)# ca save all
    Pixfirewall(config)# write memory

Configuring the IPSEC Transform Set

After IKE Phase 1 configuration is complete, a transform-set policy must be created. The transform-set defines the security policy to apply to traffic during IKE Phase 2:

    Pixfirewall(config)# crypto ipsec transform-set MYSET ah-sha-hmac esp-des

ESP and AH can be used concurrently. The following table details every possible option:

<table>
<thead>
<tr>
<th>AH Transforms</th>
<th>ESP Encryption Transforms</th>
<th>ESP Authentication Transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>ah-md5-hmac</td>
<td>esp-des</td>
<td>esp-md5-hmac</td>
</tr>
<tr>
<td>ah-sha-hmac</td>
<td>esp-3des</td>
<td>esp-sha-hmac</td>
</tr>
<tr>
<td></td>
<td>esp-aes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>esp-aes-192</td>
<td></td>
</tr>
<tr>
<td></td>
<td>esp-aes-256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>esp-null</td>
<td></td>
</tr>
</tbody>
</table>

Thus, if the desired configuration is ESP with 3DES for encryption and MD5 for authentication:

    Pixfirewall(config)# crypto ipsec transform-set MYSET esp-3des esp-md5-hmac
**Specifying Interesting Traffic**

The next step is to specify what traffic is **interesting**. In other words, which traffic can *initiate* the IPSEC tunnel, and can be *sent across* the tunnel. An access-list is used to identify interesting traffic:

```plaintext
Pixfirewall(config)# access-list MYLIST permit ip 192.168.1.0 255.255.255.0 10.1.1.0 255.255.255.0
```

The above access-list specifies that traffic *from* the 192.168.1.x network, destined *to* the 10.1.1.x network can both initiate and utilize the IPSEC tunnel. This access-list will be referenced later in our configuration.

Next, the PIX/ASA firewall must be instructed to *not* NAT this VPN traffic. The `nat 0` command coupled with an access-list will accomplish this:

```plaintext
Pixfirewall(config)# access-list NONAT permit ip 192.168.1.0 255.255.255.0 10.1.1.0 255.255.255.0
Pixfirewall(config)# nat (inside) 0 access-list NONAT
```

**Optional Commands**

Two additional commands may be necessary:

```plaintext
Pixfirewall(config)# sysopt connection permit-ipsec
Pixfirewall(config)# route outside 10.1.1.0 255.255.255.0 77.1.1.1
```

The `sysopt connection permit-ipsec` command instructs the PIX to bypass any access-lists applied to interfaces, for traffic being sent across the IPSEC tunnel. Otherwise, if PIXFIREWALL had an access-list applied for incoming traffic on its outside interface, any traffic from 10.1.1.x/24 would be dropped unless explicitly permitted.

The `route` command adjusts the PIX’s routing table, so that any traffic destined to 10.1.1.x/24 is sent across the IPSEC tunnel to the 77.1.1.1 peer.
**Configuring the IPSEC Crypto Map**

The final step is to create a crypto map, which references all previously configured IPSEC SA parameters (including the access-list for interesting traffic, the SA peer, and the IKE transform-set).

```bash
Pixfirewall(config)# crypto map MYMAP 1 ipsec-isakmp
Pixfirewall(config)# crypto map MYMAP 1 match address MYLIST
Pixfirewall(config)# crypto map MYMAP 1 set peer 77.1.1.1
Pixfirewall(config)# crypto map MYMAP 1 set transform-set MYSET
Pixfirewall(config)# crypto map MYMAP 1 set security-association lifetime seconds 8000
```

The crypto map must be applied to an interface, which will allow the IPSEC communication process to begin:

```bash
Pixfirewall(config)# crypto map MYMAP interface outside
```

Only one crypto map is allowed per interface. If multiple VPN tunnels will be terminating on a single interface, they can be separated using the sequence number within a single crypto map:

```bash
Pixfirewall(config)# crypto map MYMAP 1 match address MYLIST
Pixfirewall(config)# crypto map MYMAP 1 set peer 77.1.1.1
Pixfirewall(config)# crypto map MYMAP 1 set transform-set MYSET
Pixfirewall(config)# crypto map MYMAP 5 match address ANOTHERLIST
Pixfirewall(config)# crypto map MYMAP 5 set peer 88.1.1.1
Pixfirewall(config)# crypto map MYMAP 5 set transform-set ANOTHERSET
```

To view a configured crypto map:

```bash
Pixfirewall(config)# show crypto map
```
**Troubleshooting IPSEC Tunnels**

Various commands can be used to troubleshoot both IPSEC and ISAKMP:

```
Pixfirewall# show isakmp
Pixfirewall# show isakmp policy
Pixfirewall# show crypto ipsec transform-set
Pixfirewall# show crypto map
Pixfirewall# show crypto ipsec sa
Pixfirewall# debug crypto isakmp
Pixfirewall# debug crypto ipsec
```

To manually tear down an ISAKMP or IPSEC SA:

```
Pixfirewall# clear crypto isakmp sa
Pixfirewall# clear crypto ipsec sa
```

The following is an example of ISAKMP/IPSEC debug output:

```
7w4d: ISAKMP (0:2): beginning Quick Mode exchange, M-ID of 2023464223
7w4d: ISAKMP (0:2): sending packet to 77.1.1.1 (I) QM_IDLE
7w4d: ISAKMP (0:2): Node 2023464223, Input = IKE_MESG_INTERNAL, IKE_INIT_QM
    Old State = IKE_QM_READY  New State = IKE_QM_I_QM1

7w4d: ISAKMP (0:2): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
    Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE

7w4d: ISAKMP (0:2): received packet from 77.1.1.1 (I) QM_IDLE
7w4d: ISAKMP (0:2): processing HASH payload. message ID = -1688543668
7w4d: ISAKMP (0:2): processing NOTIFY PROPOSAL_NOT_CHOSEN protocol 0
    spi 0, message ID = -1683843668, sa = 827636AFC

7w4d: ISAKMP (0:2): deleting node -1688543668 error FALSE reason
    "informational (in) state 1"
7w4d: ISAKMP (0:2): Input = IKE_MESG_FROM_PEER, IKE_INFO_NOTIFY
    Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE
```

The above debug output is indicating that IKE Phase 1 is completing, but Phase 2 is failing due to an ESP Hash mismatch between the SA peers.