



**HEANET**  
**NETWORKING CONFERENCE**  
**LIMERICK 2004**

**SECURITY WORKSHOP**  
**“IPSEC-BASED VPN CONFIGURATION”**

**Warren Daly – HEAnet Ltd**  
**Jarek Woznica – HEAnet Ltd**

Thank you to Cisco Systems Ireland & LanComms for loan of Cisco PIX firewalls

<b>1. IPSEC INTRODUCTION.....</b>	<b>4</b>
1.1. IPSEC OVERVIEW.....	4
1.1.1. IPsec Protocols.....	5
1.1.2. IKE and IPsec.....	5
1.1.3. Tunnel and Transport Modes.....	6
1.1.4. Transforms and Transform sets.....	8
1.1.5. Transform Sets.....	8
1.1.6. IPsec Security Associations (SA).....	8
1.2. HOW IPSEC WORKS – IPSEC’S OPERATION.....	9
<b>2. CONFIGURING IPSEC ON CISCO IOS ROUTERS – PRE-SHARED KEYS.....</b>	<b>10</b>
2.1. TASK 1 – PREPARING FOR IKE AND IPSEC.....	10
2.1.1. Determining IKE (Phase One) and IPsec (Phase Two) Policies.....	10
2.1.2. Checking the current configuration.....	12
2.1.3. Ensuring that the network works.....	12
2.1.4. Ensuring that existing ACLs are compatible with IPsec.....	12
2.2. TASK 2 – CONFIGURING IKE (PHASE ONE).....	13
2.2.1. Enabling IKE.....	13
2.2.2. Creating IKE Policies.....	13
2.2.3. Configuring Preshared Keys.....	14
2.2.4. Verifying IKE configuration.....	14
2.3. TASK 3 – CONFIGURING IPSEC (PHASE TWO).....	14
2.3.1. Configuring the transform sets.....	15
2.3.2. Configuring Global IPsec SA Lifetimes.....	15
2.3.3. Creating Crypto ACLs.....	15
2.3.4. Creating Crypto Maps.....	16
2.3.5. Applying Crypto Maps to the interfaces.....	17
2.4. TASK 4 – VERIFYING IPSEC.....	17
2.4.1. ISAKMP show command.....	17
2.4.2. IPsec show commands.....	17
2.4.3. Monitoring and managing IKE and IPsec.....	17
2.4.4. IPsec debug commands.....	17
<b>3. CONFIGURING IPSEC ON CISCO PIX FIREWALLS – PRE-SHARED KEYS.....</b>	<b>19</b>
3.1. TASK 1 – PREPARING FOR IKE AND IPSEC.....	19
3.1.1. Determining IKE (IKE Phase One) policy.....	19
3.1.2. Determining IPsec (IKE Phase Two) policy.....	19
3.1.3. Checking the current configuration.....	19
3.1.4. Ensuring the compatibility of existing ACLs with the IPsec protocols.....	19
3.2. TASK 2 – CONFIGURING IKE (PHASE ONE).....	19
3.2.1. Enabling IKE.....	20
3.2.2. Creating IKE Policies.....	20
3.2.3. Configuring Pre-shared Keys.....	20
3.2.4. Verifying IKE configuration.....	20
3.3. TASK 3 – CONFIGURING IPSEC (IKE PHASE TWO).....	21
3.3.1. Configuring the transform sets.....	21
3.3.2. Configuring Global IPsec SA Lifetimes.....	21

3.3.3	Creating Crypto ACLs.....	21
3.3.4	Creating Crypto Maps.....	22
3.3.5	Applying Crypto Maps to the interfaces.....	23
3.4.	<b>TASK 4 – VERIFYING IPSEC.....</b>	<b>23</b>
3.4.1	ISAKMP show commands.....	23
3.4.2	IPSec show commands.....	23
3.4.3	Monitoring and managing IKE and IPSec.....	23
3.4.4	IPSec debug commands.....	23
<b>4.</b>	<b>TROUBLESHOOTING IPSEC.....</b>	<b>24</b>
4.1.	CISCO IOS ROUTERS.....	24
4.2.	CISCO PIX FIREWALLS.....	27
<b>5.</b>	<b>HEANET CONFERENCE – WORKSHOP EXERCISE.....</b>	<b>30</b>
5.1.	IPSEC CONFIGURATION ON CISCO IOS ROUTERS.....	30
5.1.1	Task 1) Preparation for IKE and IPSEC.....	30
5.1.2	Task 2) IKE configuration.....	30
5.1.3	Task 3) IPSec configuration.....	30
5.1.4	Task 4) Verifying IPSec.....	31
5.2.	IPSEC CONFIGURATION ON CISCO PIX FIREWALLS.....	35
5.2.1	Task 1) Preparation for IKE and IPSEC.....	35
5.2.2	Task 2) IKE configuration.....	35
5.2.3	Task 3) IPSec configuration.....	35
5.2.4	Task 4) Verifying IPSec.....	36
<b>6.</b>	<b>MEMORY AND CPU CONSIDERATIONS.....</b>	<b>37</b>
<b>7.</b>	<b>GLOSSARY.....</b>	<b>38</b>
<b>8.</b>	<b>CISCO ROUTERS AND PIX FIREWALLS WORKSHOP CONFIGURATION OUTPUTS.....</b>	<b>42</b>
8.1.	BASIC CONFIGURATION – NO IPSEC.....	42
8.1.1	Central Router.....	42
8.1.2	Branch router.....	43
8.1.3	Cisco PIX firewall.....	44
8.2.	IPSEC CONFIGURATION.....	46
8.2.1	Central Router.....	46
8.2.2	Branch router.....	47
8.2.3	Cisco PIX firewall.....	48
<b>9.</b>	<b>NETWORK DIAGRAMS.....</b>	<b>50</b>

## 1. IPSEC INTRODUCTION

IPSec's main design goals are to provide the follow functionality:

- **Data confidentiality** – data is encrypted before being transmitted, so nobody except the communicating parties can read it.
- **Data integrity** – each peer can determine if a received packet was changed during transit.
- **Data origin authentication** – as an additional feature of data integrity service, the receiver can also check the identity of a packet's sender.
- **Anti-replay** – the receiver can detect and reject replayed packets, protecting it from spoofing and man-in-the-middle attacks.

The aim of this document is to describe and explain the components and configuration methods of IPSec technology. This document does not cover all the aspects and possible scenarios of IPSec services and readers are encouraged to search for more information sources, if necessary.

### 1.1. IPSec Overview

IPSec is a suite of cryptography-based protection services and security protocols. Because it requires no changes to programs or protocols, it can be easily deployed for existing networks.

IPSec can be used to protect the integrity and confidentiality of corporate data, to authorize specific users to utilize corporate resources, to authenticate machines, and to avoid replay of a network operating session. It does this by encrypting communications between endpoints using well-known encryption algorithms, and including mechanisms for authentication of datagrams and privacy of the datagram payload.

Encryption is determined by the IPSec Security Association, or SA. A security association is a combination of a destination address, a security protocol, and a unique identification value, called a Security Parameters Index (SPI). The available encryptions include:

- Data Encryption Standard (**DES**), which uses 56-bit key;
- Triple DES (**3DES**), which uses two 56-bit keys and is designed for high-security environments in North America.

According to Douglas Comer, author of *Internetworking with TCP/IP: Principles, Protocols, and Architectures*, information or data security is comprised of these aspects:

- **Data integrity** – protecting information from tampering;

- **Data availability** – information should be available to authorized users;
- **Privacy or Confidentiality** – information should be readable only by authorized users;
- **Authorization** – controlling who is authorized for an item of information;
- **Authentication** – two entities should be able to validate one another;
- **Replay avoidance** – a retransmitted copy of data should not be accepted later, because it may come from a “bad guy”.

### 1.1.1IPSec Protocols

Two primary types of IPSec protocols exist:

- IP Type 50 **Encapsulating Security Payload (ESP)**: ESP provides authentication and encryption;
- IP Type 51 **Authentication Header (AH)**: AH provides authentication but not encryption.

ESP encrypts the TCP or UDP header along with the payload data. Whether IPSec modifies the original IP header depends on the **IPSec mode**. IPSec supports transport or tunnel mode, both of which can use either ESP or AH packets.

- Transport mode secures packets between two endpoints, typically in a client-to-gateway scenario, and leaves the original IP header unchanged;
- Tunnel mode encapsulates the IP header and payload into a new IPSec packet for transfer between two endpoints, typically two IPSec gateway devices.

In either mode, both IPSec endpoint devices use Internet Key Exchange (IKE) to negotiate authentication and use encryption if necessary to support ESP.

### 1.1.2IKE and IPSec

The Internet Key Exchange (IKE) protocol is a key management protocol standard that is used in conjunction with the IPSec standard.

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

IKE is a hybrid protocol that implements the **Oakley** key exchange and **Skeme** key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework.

ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.

### 1.1.3 Tunnel and Transport Modes

The order of the headers is as follows:

- **Transport Mode:** IP header, IPSec headers (AH and/or ESP), IP payload (including transport header);
- **Tunnel Mode:** New IP header, IPSec headers (AH and/or ESP), old IP header, IP payload.

AH is a protocol that provides **authentication** of either all or part of the contents of a datagram through the addition of a **header** that is calculated based on the values in the datagram. What parts of the datagram are used for the calculation, and the placement of the header, depends on the mode (tunnel or transport) and the version of IP (IPv4 or IPv6).

The operation of the AH protocol is simple— It can be considered analogous to the algorithms used to calculate checksums or perform CRC checks for error detection. This computed result is transmitted along with the original data to the destination, which repeats the calculation and discards the message if any discrepancy is found between its calculation and the one done by the source.

This is the same idea behind AH, except that instead of using a simple algorithm known to everyone, a special hashing algorithm is used and a specific key known only to the source and the destination. A Security Association between two devices is set up that specifies these particulars so that the source and destination know how to perform the computation but nobody else can. On the source device, AH performs the computation and puts the result (called the **Integrity Check Value** or **ICV**) into a special header with other fields for transmission. The destination device does the same calculation using the key the two devices share, which enables it to see immediately if any of the fields in the original datagram were modified (either due to error or malice).

The IPSec Authentication Header (AH) provides integrity authentication services to IPSec-capable devices, so they can verify that messages are received intact from other devices. For many applications, however, this is only one piece of the puzzle. Not only protection against intermediate devices changing the datagrams is required, but also protection against them examining their contents as well. For private communications, AH is not enough; **Encapsulating Security Payload (ESP)** protocol needs to be used.

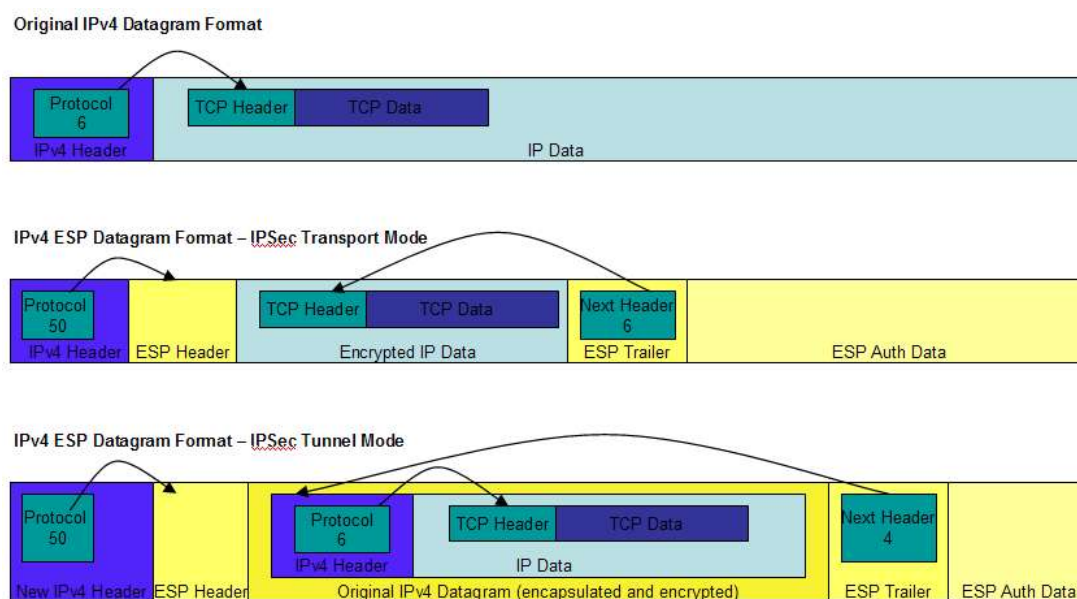
The main job of ESP is to provide the privacy sought for IP datagrams by **encrypting** them. An encryption algorithm combines the data in the datagram with a key to transform it into an encrypted form. This is then repackaged using a special format, seen shortly, and transmitted to the destination, which decrypts it using the same algorithm. Encapsulating Security Payload supports its own authentication scheme, like that used in AH, or can be used in conjunction with AH.

## Encapsulating Security Payload Fields

ESP has several fields that are the same as those used in AH, but packages its fields in a very different way. Instead of having just a header, it divides its fields into three components:

- **ESP Header:** This contains two fields, the **SPI** and **Sequence Number**, and comes before the encrypted data.
- **ESP Trailer:** This section is placed after the encrypted data. It contains padding that is used to align the encrypted data, through a **Padding** and **Pad Length** field. Interestingly, it also contains the **Next Header** field for ESP;
- **ESP Authentication Data:** This field contains an **Integrity Check Value (ICV)**, computed in a manner similar to how the AH protocol works, for when ESP's optional authentication feature is used.

There are two reasons why these fields are broken into pieces like this. The first is that some encryption algorithms require the data to be encrypted to have a certain block size, and so padding must appear after the data and not before it. That's why padding appears in the ESP Trailer. The second is that the **ESP Authentication Data** appears separately because it is used to authenticate the rest of the encrypted datagram **after** encryption. This means it cannot appear in the ESP Header or ESP Trailer.



**Fig 1.** IPv4 Datagram Format with IPsec ESP

#### 1.1.4 Transforms and Transform sets

##### Transforms

An IPSec **transform** specifies a single IPSec security protocol (either AH or ESP) with its corresponding security algorithms and mode. Example transforms include the following:

- The AH protocol with the HMAC with MD5 authentication algorithm in tunnel mode is used for authentication;
- The ESP protocol with the triple DES (3DES) encryption algorithm in transport mode is used for confidentiality of data.

The ESP protocol with the Triple DES encryption algorithm and the HMAC with SHA-1 authentication algorithm in tunnel mode is used for authentication and confidentiality in this workshop.

##### 1.1.5 Transform Sets

A **transform set** is a combination of individual IPSec transforms designed to enact a specific security policy for traffic. During the ISAKMP IPSec security association negotiation that occurs in IKE phase 2 **quick mode**, the peers agree to use a particular transform set for protecting a particular data flow. Transform sets combine the following IPSec factors:

- Mechanism for payload authentication – **AH transform**
- Mechanism for payload encryption – **ESP transform**
- IPSec mode (transport versus tunnel)

Transform sets equal a combination of an AH transform, plus an ESP transform, plus the IPSec mode (either tunnel or transport mode).

##### 1.1.6 IPSec Security Associations (SA)

The concept of a **security association (SA)** is fundamental to IPSec. An SA is a relationship between two or more entities that describes how the entities will use security services to communicate securely. IPSec provides many options for performing network encryption and authentication.

Each IPSec connection can provide encryption, integrity, authenticity, or all three services.

When the security service is determined, the two IPSec peers must determine exactly which algorithms to use (for example, DES or 3DES for encryption; MD5 or SHA-1 for integrity). After deciding on the algorithms, the two devices must share session keys. As can be seen, there is quite a bit of information to manage. The security association is the method that IPSec uses to track all the particulars concerning a given IPSec communication session

A separate pair of IPSec SAs is set up for AH and ESP transform. Each IPSec peer

agrees to set up SAs consisting of policy parameters to be used during the IPSec session. The SAs are unidirectional for IPSec, so that peer 1 will offer peer 2 a policy. If peer 2 accepts this policy, it will send that policy back to peer 1. This establishes two one-way SAs between the peers. Two-way communication consists of two SAs, one for each direction.

Each SA consists of values such as destination address, a **security parameter index (SPI)**, IPSec transforms used for that session, security keys, and additional attributes such as IPSec lifetime. The SAs in each peer have unique SPI values that will be recorded in the security parameter database on each device.

## 1.2. How IPSec works – IPSec's Operation

IPSec's operation can be broken down into five main steps, as follows:

**Step 1) Interesting traffic initiates the IPSec process** – the Crypto ACLs define the interesting traffic – Crypto ACLs are explained in section [2.3.4. Create Crypto ACLs](#);

**Step 2) IKE Phase One** – IKE authenticates IPSec peers and negotiates IKE SAs during this phase, setting up a secure channel for negotiating IPSec SAs in Phase Two, which is described in section [2.2. Configure IKE \(Phase One\)](#);

**Step 3) IKE Phase Two** – IKE negotiates IPSec SA parameters and sets up matching IPSec SAs in the peers – section [2.3.s Configure IPSec \(Phase Two\)](#)

**Step 4) Data transfer** – Data is transferred between IPSec peers based on the IPSec parameters and keys stored in the SA database;

**Step 5) IPSec tunnel termination** – IPSec SAs terminate through deletion or by timing out;

## **2.CONFIGURING IPSEC ON CISCO IOS ROUTERS – PRE-SHARED KEYS**

### **2.1.Task 1 – Preparing for IKE and IPsec**

The proper preparation for IPsec services configuration is a key factor in successful implementation of IPsec-based VPN networks.

IPsec services configuration is a complicated and complex task therefore planning well in advance helps to avoiding configuration problems and other issues.

The sections below describe the necessary steps to follow in order to prepare for IPsec deployment.

#### **2.1.1Determining IKE (Phase One) and IPsec (Phase Two) Policies**

Before any configuration commands are entered into Cisco router, decisions have to be made how the IPsec-based VPN network will be deployed and implemented.

The questions to be answered here are:

- Determine the key distribution method
  - Manual – used in the networks with small number of IPsec peers – this is the method described in this section;
  - CA server – used in the networks with large number of IPsec hosts, where scalability is required;
- Determine the authentication method
  - Pre-shared keys
  - RSA nonces
  - RSA signatures
- Identify IPsec peer's IP addresses (and hosts names if required) - these details will be used to configure IKE and IPsec parameters;
- Determine IKE (ISAKMP) – Phase One – policy including:
  - Encryption algorithm
  - Hash algorithm
  - IKE SA lifetime
  - Diffie-Hellman group

This policy defines a combination of security parameters used during the ISAKMP negotiation. The IKE policy parameters have to match in order for IKE SA to be established.

The table below lists IKE Phase One possible policy settings along with their default settings on the Cisco router:

Parameter	Accepted Values	Keyword	Default
Message encryption algorithm	DES 3DES	<b>des</b> <b>3des</b>	DES
Message integrity (hash) algorithm	SHA-1 (HMAC variant) MD5 (HMAC variant)	<b>sha</b> <b>md5</b>	SHA-1
Peer authentication method	Preshared keys RSA encrypted nonces RSA signatures	<b>pre-share</b> <b>rsa-encr</b> <b>rsa-sig</b>	RSA signatures
Key exchange parameters (D-H group identifier)	768-bit Diffie-Hellman 1024-bit Diffie-Hellman	<b>1</b> <b>2</b>	768-bit D-H
ISAKMP-established lifetime	Number of seconds	-	86400 secs (one day)

- Determine IPsec – Phase Two – policy including:
  - IPsec algorithms and parameters – it is important to take into consideration the trade-off between the high performance and stronger security;
  - Select transforms and transform sets;
  - Decide what type of traffic (which networks, hosts, applications etc) is protected by IPsec;

The tables below lists the IPsec transforms supported by Cisco IOS

**Authentication Header (AH) transforms**

ah-md4-hmac	AH-MD5-HMAC transform
ah-sha-hmac	AH-SHA-HMAC transform
ah-rfc1828	AH-MD5 transform (RFC1828) used with older IPsec implementations

**Please note:**

AH is rarely used since authentication is now available with the esp-sha-hmac and esp-md5-hmac transforms.

**Encapsulating Security Payload (ESP) transforms:**

esp-des	ESP transform using DES cipher (56 bit)
esp-3des	ESP transform using 3DES cipher (168 bit)
esp-md5-hmac	ESP transform using HMAC-MD5 authentication used with esp-des and esp-3des transforms
esp-sha-hmac	ESP transform using HMAC-SHA authentication used with esp-des and esp-3des transforms
esp-null	ESP transform with no cipher used with esp-md5-hmac and esp-sha-hmac if authentication and no encryption is required – not recommended – no protection for data flow
esp-rfc1829	ESP-DES-CBC transform used with older IPSec implementations

During the transform set configuration the Cisco IOS prevents to entering the incorrect combinations of transforms, e.g. once an AH transform is chosen, it does not allow to configure yet another one in the same transform set.

Because choosing the transform sets combination can be complex, the following tips might be helpful:

- If data confidentiality is required, ESP encryption should be used, either esp-des (strong) or esp-3des (stronger);
- If authentication is required, either ESP authentication transform: esp-md5-hmac or esp-sha-hmac, or AH authentication: ah-md5-hmac or ah-sha-hmac can be used

**2.1.2 Checking the current configuration**

It is always useful to make sure whether the Cisco router where IPSec services are about to be configured has already IPSec policy defined on it, which may be useful or may interfere with the planned IPSec policy.

The commands below help to clarify present IPSec services configuration:

- **show running-config**
- **show crypto isakmp policy** – to verify IKE policy configuration (if any)
- **show crypto map** – to verify IPSec policy configuration (if any)
- **show crypto ipsec transform-set** – to view configured transform sets

**2.1.3 Ensuring that the network works**

Before IPSec services are configured the basic connectivity (layer 3) between the peers is required.

Not only simple ICMP protocol (ping) should be verified, but also any other protocols and applications connectivity, which are to be protected by IPSec.

It is much more difficult to troubleshoot the basic connectivity while IPSec is activated.

### 2.1.4 Ensuring that existing ACLs are compatible with IPsec

ACLs configured on the routers – specifically perimeter routers, which implement a restrictive security policy – may block IPsec traffic.

It is essential to make sure the ACLs permit ISAKMP (UDP port 500), Encapsulating Payload Security (ESP – IP protocol number 50) and Authentication Header (AH – IP protocol 51).

The ACLs configured on the router can be simply verified by **show access-lists** command.

## 2.2. Task 2 – Configuring IKE (Phase One)

The aim of this task is to configure IKE parameters using information gathered in the earlier task. This section presents the steps necessary in configuring IKE policies.

### 2.2.1 Enabling IKE

The first step is to enable ISAKMP (IKE). IKE is globally enabled with the following command: **crypto isakmp enable**. On the Cisco routers, ISAKMP is enabled by default.

### 2.2.2 Creating IKE Policies

The next step is to configure the IKE policies, which are used to establish ISAKMP peering between two IPsec endpoints.

It is possible to configure more than one IKE policy, which then are negotiated between IPsec peers – the matching one is chosen. A priority number (1 – 10000) is used in order to specify the hierarchy of IKE policies – the lower number the higher priority. Routers start looking the match from the policy with the highest priority. It is recommended to assign the most secure policy the highest priority so it is attempted and matched first.

During the ISAKMP negotiation process (IKE Phase One main mode), ISAKMP looks for an ISAKMP policy that is the same on both peers. Peer initiating the negotiation sends all its configured IKE policies to the remote peer, which then tries to find a match with its policies (starting from the policy configured with the highest priority).

A match means that both policies have the same encryption, hash authentication and Diffie-Hellman group configured and the lifetime is equal or higher than the lifetime specified in the remote peer's policy.

If no matching policy is found ISAKMP refuses negotiation and IPsec is not established.

The command **crypto isakmp policy *priority*** invokes IKE policy configuration command mode (*config-iskamp*).

Below are the options available in config-iskamp mode:

ISAKMP commands:

```
authentication {rsa-sig | rsa-encr | preshare}
default
encryption {des | 3des}
```

exit  
 group  
 hash {md5 | sha}  
 lifetime seconds  
 no

The table below provides the explanation for some of the keywords and options:

Keyword	Accepted Values	Default value	Description
<b>des</b>	DES	DES	Message encryption algorithm
<b>3des</b>	3DES		
<b>sha</b>	SHA-1 (HMAC variant)	SHA-1	Message integrity (hash) algorithm
<b>md5</b>	MD5 (HMAC variant)		
<b>pre-share</b>	Preshared keys	RSA signatures	Peer authentication method
<b>rsa-encr</b>	RSA encrypted nonces		
<b>rsa-sig</b>	RSA signatures		
<b>1</b>	768-bit Diffie-Hellman	768-bit D-H	Key exchange parameters
<b>2</b>	1024-bit Diffie-Hellman		(D-H group identifier)
<b>-</b>	Number of seconds	86400 secs (one day)	ISAKMP-established lifetime
<b>exit</b>			Exits the config-iskamp mode

### 2.2.3 Configuring Preshared Keys

Another step is to configure the preshared keys along with the peer's identity – either router's IP address (default) or hostname – which are used to authenticate the peers to each other during ISAKMP negotiation process.

To change the identity method – either IP address or hostname – the following command is used: **crypto isakmp identity {address | hostname}**. Again, IP address is the default identity.

If the hostname identity is used, either DNS server must be available or hostname for the remote peer has to be configured using the following command:

**ip host** *hostname IP\_address*

The next command is used to configure the preshared authentication key:

**crypto isakmp key** *keystring address peer-address* => uses IP address as identity

**crypto isakmp key** *keystring hostname peer-address* => uses hostname as identity

### 2.2.4 Verifying IKE configuration

To verify IKE configuration command: **show crypto isakmp policy** is used.

### 2.3.Task 3 – Configuring IPSec (Phase Two)

This task explains the process of configuring IPSec parameters, again using information gathered in Task 1.

#### 2.3.1Configuring the transform sets

The first step is to configure the Transform Set Suites. As explained earlier, the transform set is a combination of individual IPSec transforms providing specific security policy for the traffic.

During the negotiation process occurring in IKE Phase Two, quick mode, the peers agree on particular transform set for protecting a particular data flow. The transform sets combine the following IPSec factors:

- Mechanism for payload authentication – AH;
- Mechanism for payload encryption – ESP;
- IPSec mode – transport versus tunnel;

Global configuration command defines transform sets – up to three transforms can be specified:

```
crypto ipsec transform-set transform_set_name transform1 [transform2 [transform3]]
```

#### 2.3.2Configuring Global IPSec SA Lifetimes

Optionally, if defaults are not used, the IPSec SA lifetime may be configured using the following global configuration command:

```
crypto ipsec security-association lifetime {seconds seconds | kilobytes kilobytes}
```

The defaults are as follows:

- 3600 seconds (one hour)
- 4 608 000 kilobytes

The IPSec SA Lifetimes have following characteristics:

- Global IPSec SA lifetime is used by all crypto maps;
- IPSec SA lifetimes are negotiated during IKE Phase Two;
- Lifetimes can be optionally configured in Crypto Maps;
- Crypto Map IPSec SA lifetimes override Global IPSec SA lifetimes;

#### 2.3.3Creating Crypto ACLs

The next step involves defining Crypto ACLs. The meaning of Crypto ACLs differs from the ordinary ACLs in that they specify the traffic, which must be protected and the traffic, which need not be protected.

In Crypto ACLs **permit** statement defines all the traffic to be protected, and **deny** statement defines the traffic not protected – not protected traffic is forwarded in a usual way.

The Crypto ACLs are the extended ACLs and are applied to the outbound interface.

It is important to note that any unprotected traffic, which matches the **permit** statement, will be silently dropped, because router expects such traffic to be protected by IPSec.

Below is a general extended ACL command syntax used to define Crypto ACLs:

**access-list** *acl-num* {**permit** | **deny**} *protocol source src-wildcard destination dst-wildcard*

It is not recommended to use **any** keyword to specify source or destination addresses – the use of **permit any any** statement is strongly discouraged as this will cause all outbound traffic to be protected and will require protection on all incoming traffic. This may result in dropping all inbound not protected packets, including packets for routing protocol such as NTP, echo, echo response, etc.

It is recommended (if not necessary) to configure symmetrical (mirror images) Crypto ACLs on both peers – for example the source criteria on Router1 should be the same as the destination criteria on Router2, and the destination criteria on Router1 should be the same as the source criteria on Router2:

Router1

```
access-list 110 permit ip 192.162.10.0 0.0.0.255 192.168.1.0 0.0.0.255
```

Router2

```
access-list 110 permit ip 192.168.1.0 0.0.0.255 192.162.10.0 0.0.0.255
```

### 2.3.4 Creating Crypto Maps

The Crypto Map entries tie together all the various parts configured for IPSec so far, namely:

- which traffic should be protected by IPSec (Crypto ACLs);
- where IPSec-protected traffic should be sent – IPSec peer;
- The local address to be used for the IPSec traffic
- What IPSec security should be applied to this traffic – transform sets

It is possible to create more than one Crypto Map, which are differentiated by sequence number – the lower is the sequence number the higher is the priority.

Only one Crypto Map can be applied to the outbound interface – see next section.

The reason to set up more than one Crypto Maps may be for example requirement to define different IPSec security to different types of traffic – the different type of traffic should be defined in two separate Crypto ACLs, which then are used by two separate Crypto Maps.

The following global configuration command creates the Crypto Map and enters the Crypto Map configuration mode:

**crypto map** *map-name sequence-number ipsec-isakmp*

Within the crypto map configuration mode following option are available:

- **set** – used with the **peer**, **pfs**, **transform-set** and **security-association** commands;

- **peer [hostname | address]** – specifies the IPsec peer;
- **pfs [group1 | group2]** – Specifies D-H algorithm group1 or group2;
- **transform-set** – specifies the list of transform-sets in priority order – up to six transform-sets can be specified;
- **security-association lifetime** – sets the lifetime (seconds or kilobytes) of SA per crypto map;
- **match address [access-list-id | name]** – identifies the Crypto ACL – the value should match the previously defined ACL's number or name argument;

### 2.3.5 Applying Crypto Maps to the interfaces

The last step in configuring the IPsec is to apply the Crypto Map defined in the previous step to an interface – as already mentioned, only one crypto map can be applied to the interface.

The Crypto Map is applied to the outbound interface within interface configuration mode with this command:

```
crypto map map-name
```

### 2.4.Task 4 – Verifying IPsec

Once all the parameter of IKE (Phase One) and IPsec (Phase Two) are defined and configured it is important to verify the configuration. There is a set of various commands available, which help to validate the correct IPsec configuration.

#### 2.4.1 ISAKMP show command

To check the ISAKMP configuration the following command can be used:

- **show crypto map isakmp policy** – displays configured IKE policies;

#### 2.4.2 IPsec show commands

The IPsec configuration is verified with these commands:

- **show crypto ipsec transform-set** – displays configured IPsec transform sets;
- **show crypto ipsec sa** – displays the state of IPsec SAs;
- **show crypto map** – displays configured crypto maps;

#### 2.4.3 Monitoring and managing IKE and IPsec

**show isakmp sa** – displays the current status of ISAKMP SAs;

**show crypto ipsec sa** – displays the current status of IPsec SAs – useful to make sure the traffic is being encrypted;

**clear isakmp** – clears ISAKMP SAs;

**clear crypto ipsec sa** – clears IPsec SAs;

#### **2.4.4IPSec debug commands**

To debug IKE and IPSec traffic commands below are useful:

- **debug crypto ipsec**
- **debug crypto isakmp**

### **3.CONFIGURING IPSEC ON CISCO PIX FIREWALLS – PRE-SHARED KEYS**

This section covers IPsec services configuration on the Cisco PIX devices. The concept and principles of configuring IPsec services on Cisco PIX firewalls is very similar to the principles of configuring IPsec on Cisco routers described already in detail in previous section.

Therefore, in this part of the manual the tasks and steps involved during the IPsec services configuration are explained in less detail, while all the differences of configuring IPsec on Cisco PIX devices will be described in detail.

#### **3.1.Task 1 – Preparing for IKE and IPsec**

As mentioned already, the first task is one of the most important – advanced preparations before implementing IPsec is vital and will help avoiding problems during the actual configuration of IPsec services.

This task involves the following steps:

##### **3.1.1Determining IKE (IKE Phase One) policy**

Information collected during this step help on deciding upon IKE Phase One policy.

##### **3.1.2Determining IPsec (IKE Phase Two) policy**

During this step IPsec peers details are identified including IP addresses, IPsec modes, encryption levels, etc – Crypto Maps are configured based on the information gathered in this step;

##### **3.1.3Checking the current configuration**

To ensure the new, IPsec configuration will not interfere with the existing one, it is crucial to check the present configuration of the PIX device. The following commands can be used in order to verify the PIX device configuration:

**write terminal**

**show isakmp [policy]**

**show crypto map**

##### **3.1.4Ensuring the compatibility of existing ACLs with the IPsec protocols**

As already mentioned the UDP port 500, and IP protocols 50 (ESP) and 51 (AH) should not be blocked by existing ACLs in order for IPsec services working - it is crucial to make sure that perimeter routers, which are usually configured with the highest level of security do not deny traffic on these ports and protocols.

#### **3.2.Task 2 – Configuring IKE (Phase One)**

Once all the information is gathered, the IKE Phase One can be configured.

### 3.2.1 Enabling IKE

First ISAKMP has to be enabled by typing this command:

**enable isakmp interface\_name** – the interface is the interface terminating IPSec tunnel – usually the outside interface;

### 3.2.2 Creating IKE Policies

The next step is the configuration of IKE policies, which are used to establish ISAKMP peering between two IPSec endpoints. Below the commands necessary to define IKE policy are listed:

**iskamp policy priority** – defines the policy and its priority;

**iskamp policy priority encryption [des | 3des]** – defines encryption algorithm – default is des;

**iskamp policy priority hash [md5 | sha]** – defines the hash algorithm;

**iskamp policy priority authentication [pre-share | rsa-sig]** – defines the method of peers authentication – default is sha;

**iskamp policy priority group 1 | 2** – defines Diffie-Hellman algorithm group – default is group 1;

**iskamp policy priority lifetime seconds** – defines the IKE SA's lifetime – default is 86400 seconds (one day);

### 3.2.3 Configuring Pre-shared Keys

Peers authenticate each other during ISAKMP negotiations using the pre-shared key and ISAKMP identity – either IP address or hostname.

The IP address is the default identity, if hostname identity is chosen either DNS server has to be available or name-to-address mapping has to be specified.

The command to decide upon identity is:

**isakmp identity {address | hostname}**

**name ip\_address** – specifies host-to-name mapping

To configure the pre-shared key this command is used:

**isakmp key keystring address peer-address [netmask]**

### 3.2.4 Verifying IKE configuration

Once IKE policy is configured it shall be verified, using the following commands:

**write terminal** – shows the current, running configuration;

**show isakmp policy** – displays configured IKE policies;

**show isakmp** – display IKE policies again – just in different format – similar to write terminal;

### 3.3.Task 3 – Configuring IPsec (IKE Phase Two)

The next major task is configuring IPsec parameters – the steps below describe all the aspects of configuring IKE Phase Two parameters.

#### 3.3.1Configuring the transform sets

A transform set is a combination of individual transforms used to authenticate and/or encrypt specific traffic – it applies a chosen security policy to the traffic and has to be agreed between two IPsec peers. Transform sets define the following:

- Mechanism for payload authentication – AH;
- Mechanism for payload encryption – ESP;
- IPsec mode – transport versus tunnel;

Transform sets are configured with the following global configuration command:

```
crypto ipsec transform-set transform_set_name transform1 [transform2 [transform3]]
```

As in Cisco IOS IPsec, up to three transforms can be specified.

#### 3.3.2Configuring Global IPsec SA Lifetimes

Optionally, if defaults are not used, the IPsec SA lifetime may be configured using the following global configuration command:

```
crypto ipsec security-association lifetime {seconds seconds | kilobytes kilobytes}
```

The defaults are as follows:

- 3600 seconds (one hour)
- 4 608 000 kilobytes

The IPsec SA Lifetimes have following characteristics:

- Global IPsec SA lifetime is used by all crypto maps;
- IPsec SA lifetimes are negotiated during IKE Phase Two;
- Lifetimes can be optionally configured in Crypto Maps;
- Crypto Map IPsec SA lifetimes override Global IPsec SA lifetimes;

#### 3.3.3Creating Crypto ACLs

The Crypto ACLs on the Cisco PIX firewalls are configured in exactly the same way as they are on Cisco IOS routers.

Because Cisco PIX firewalls operate using the concept of NAT/PAT services, the first task is to correctly configure these services – however the detailed explanation of configuring NAT and PAT is beyond the scope of this document. More information is included on [www.cisco.com](http://www.cisco.com) website and in many publications available.

Examples within this document use the concept of static NAT is used and configured, as follows:

```
static (inside, outside) 192.168.30.3 192.168.60 1 netmask 255.255.255.255 0 0
```

**Once static NAT is configured, the Crypto ACL uses local global (translated, outside) IP address, and not the local inside IP address**

Again, the Crypto ACLs **permit** statement defines all the traffic to be protected, and **deny** statement defines the traffic not protected – not protected traffic is forwarded in a usual way.

It is important to note that any unprotected traffic, which matches the **permit** statement, will be silently dropped, because router expects such traffic to be protected by IPSec.

Below is a general syntax of ACL command used to define Crypto ACLs:

**access-list** *acl-num* {**permit** | **deny**} *protocol source src-mask destination dst-mask*

**Please note: Cisco PIX firewalls use network mask instead of wildcards when configuring access-lists.**

As mentioned already in earlier sections, it is encouraged to avoid using **any** keyword to specify source or destination addresses – the use of **permit any any** statement is strongly discouraged as this will cause all outbound traffic to be protected and will require protection on all incoming traffic. This may result in dropping all inbound not protected packets, including packets for routing protocol such as NTP, echo, echo response, etc.

Crypto ACLs should be symmetrical on both peers as already explained before:

PIX1

```
access-list 110 permit ip 192.162.30.0 255.255.255.0 192.168.1.0 255.255.255.0
```

PIX2

```
access-list 110 permit ip 192.168.1.0 255.255.255.0 192.162.30.0 255.255.255.0
```

### 3.3.4 Creating Crypto Maps

As already explained, Crypto Map tie up together various parts required to set up IPSec services – see the section [2.3.4 Creating Crypto Maps](#).

The Crypto Map and its parameters are configured with the following commands:

1) The Crypto Map is created:

**crypto map** *map-name sequence-number ipsec-isakmp*

2) Crypto ACL is assigned to Crypto Map:

**crypto map** *map-name sequence-number match address crypto\_acl*

3) The IPSec peer to which protected traffic is forwarded is specified:

**crypto map** *map-name sequence-number set peer hostname | ip\_address*

4) Defined Transform sets are assigned to the Crypto Map:

**crypto map** *map-name sequence-number set transform-set transform-set1 [transform-set2] [transform-set3]*

Up to nine transform sets can be specified within Crypto Map – they should be listed in order of priority, the highest priority first and the most secure transform-set should have the highest priority.

5) Optionally the IPsec SA Lifetime can be specified:

**crypto map** *map-name sequence-number* **set security-association lifetime seconds seconds | kilobytes kilobytes**

### 3.3.5 Applying Crypto Maps to the interfaces

Once Crypto Map is defined, it has to be applied to the interface – only one crypto map can be assigned to the interface, however if multiple Crypto Maps have the same name, but different sequence number, they are considered the part of the same set and all are applied to the interface.

**crypto map** *map-name sequence-number* **interface** *interface\_name*

As soon as Crypto Map is applied to the interface, SA should initialise.

## 3.4. Task 4 – Verifying IPsec

The last task of configuring IPsec services is the verification of IPsec configuration – this section summarises the methods and commands used to test and verify IPsec VPN configuration.

### 3.4.1 ISAKMP show commands

To verify IKE Phase one configuration, following commands are used:

**show isakmp** – displays configured ISAKMP policies in similar format to **write terminal (show run)** command;

**show isakmp policy** – displays default and any configured ISAKMP policies

### 3.4.2 IPsec show commands

**show access-list** – displays access-list command statements and list the hit-count of every entry – used to verify Crypto ACLs;

**show crypto map** – displays configured Crypto Maps and Crypto ACLs assigned to them;

**show crypto ipsec transform-set** – displays configured IPsec transform sets;

**show crypto ipsec security-association lifetime** – displays global IPsec SA Lifetime values;

### 3.4.3 Monitoring and managing IKE and IPsec

**show isakmp sa** – displays the current status of ISAKMP SAs;

**show crypto ipsec sa** – displays the current status of IPsec SAs – useful to make sure the traffic is being encrypted;

### 3.4.4 IPsec debug commands

**debug crypto isakmp** – displays IKE communications between the IPsec peers

**debug crypto ipsec** – displays IPsec communications between the IPsec peers;

## 4. TROUBLESHOOTING IPSEC

This section presents basic troubleshooting methods used during solving the problems with IPsec services.

### 4.1. Cisco IOS routers

The basic tools during troubleshooting IPsec services on Cisco IOS Routers are **show** and **debug** commands – they help to track down the reasons of the problems and fix them.

The **show** commands can be divided into two groups: Display show commands and Information show commands.

- **Display show commands**

These commands allow to quickly overview the current IPsec services configuration and confirm it is correct on both ends of IPsec tunnel:

#### **show crypto isakmp policy**

```
Global IKE policy
Protection suite of priority 1
  encryption algorithm: Three key triple DES
  hash algorithm:      Secure Hash Standard
  authentication method: Pre-Shared Key
  Diffie-Hellman group: #1 (768 bit)
  lifetime:           86400 seconds, no volume limit
Default protection suite
  encryption algorithm: DES - Data Encryption Standard (56
bit keys).
  hash algorithm:      Secure Hash Standard
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group: #1 (768 bit)
  lifetime:           86400 seconds, no volume limit
```

#### **show crypto isakmp key**

```
Keyring      Hostname/Address      Preshared Key
default      10.1.1.1              clsc0
```

#### **show crypto ipsec transform-set**

```
Transform set TS: { esp-3des esp-sha-hmac }
  will negotiate = { Tunnel, },
```

#### **show crypto map**

```
Crypto Map "CM" 1 ipsec-isakmp
  Peer = 10.1.1.1
  Extended IP access list 101
    access-list 101 permit ip 192.168.10.0 0.0.0.255
192.168.1.0 0.0.0.255
  Current peer: 10.1.1.1
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
```

```

Transform sets={
    TS,
}
Interfaces using crypto map CM:
    FastEthernet0/1
    
```

**show crypto map interface interface\_name**

```

Crypto Map "CM" 1 ipsec-isakmp
  Peer = 10.1.1.1
  Extended IP access list 101
    access-list 101 permit ip 192.168.10.0 0.0.0.255
192.168.1.0 0.0.0.255
  Current peer: 10.1.1.1
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    TS,
  }
Interfaces using crypto map CM:
    FastEthernet0/1
    
```

- **Information show commands**

These commands allow to view the status of established IPSec-based VPN tunnels:

show crypto engine connections active

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	FastEthernet0/1	10.1.1.2	set	HMAC_SHA+3DES_56_C	0	0
2000	FastEthernet0/1	10.1.1.2	set	HMAC_SHA+3DES_56_C		
	0	11				
2001	FastEthernet0/1	10.1.1.2	set	HMAC_SHA+3DES_56_C		
	12	0				

show crypto ipsec sa

```

interface: FastEthernet0/1
  Crypto map tag: CM, local addr. 10.1.1.2

protected vrf:
local ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
current_peer: 10.1.1.1:500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 88, #pkts encrypt: 88, #pkts digest 88
  #pkts decaps: 79, #pkts decrypt: 79, #pkts verify 79
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 4, #recv errors 0

local crypto endpt.: 10.1.1.2, remote crypto endpt.: 10.1.1.1
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/1
current outbound spi: DDD8453F

inbound esp sas:
  spi: 0xA47A5866(2759481446)
  transform: esp-3des esp-sha-hmac ,
  in use settings ={Tunnel, }
  slot: 0, conn id: 2000, flow_id: 1, crypto map: CM
  sa timing: remaining key lifetime (k/sec): (4448831/3299)
  IV size: 8 bytes
  replay detection support: Y
    
```

```

inbound ah sas:

inbound pcg sas:

outbound esp sas:
spi: 0xDDD8453F(3721938239)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2001, flow_id: 2, crypto map: CM
sa timing: remaining key lifetime (k/sec): (4448831/3297)
IV size: 8 bytes
replay detection support: Y

outbound ah sas:

outbound pcg sas:

```

Below, the usual reasons causing the problems with establishing the IPSec tunnels are listed:

- **Incompatible ISAKMP policies** – to solve the problems related with ISAKMP policies **show crypto isakmp policy** command can be used – comparing the outputs from this command helps noticing the incompatibilities – the ISAKMP policies should match;

Furthermore, **debug crypto isakmp** command can be used to troubleshoot problems with ISAKMP – the part of debug command output below displays the successful negotiation of ISAKMP parameters:

```

*Jan 11 02:42:01.517: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 1
policy
*Jan 11 02:42:01.517: ISAKMP:          encryption 3DES-CBC
*Jan 11 02:42:01.517: ISAKMP:          hash SHA
*Jan 11 02:42:01.517: ISAKMP:          default group 1
*Jan 11 02:42:01.517: ISAKMP:          auth pre-share
*Jan 11 02:42:01.517: ISAKMP:          life type in seconds
*Jan 11 02:42:01.517: ISAKMP:          life duration (VPI) of 0x0 0x1 0x51 0x80
*Jan 11 02:42:01.517: ISAKMP (0:3): atts are acceptable. Next payload is 0
! The IPSec peers found a matching ISAKMP policy

```

Another part of debug command output below displays the unsuccessful negotiation of ISAKMP parameters:

```

*Jan 11 02:42:01.517: ISAKMP (0:3): beginning Main Mode Exchange
*Jan 11 02:42:01.517: ISAKMP (0:3): sending packet to 10.1.1.1 (I) MM_NO_STATE.
*Jan 11 02:42:01.517: ISAKMP (0:3): received packet from 10.1.1.1 (I) MM_NO_STATE.
*Jan 11 02:42:01.517: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Informational mode
failed with peer at 10.1.1.1

```

- **Differing Preshared keys between the IPSec peers** – the preshared keys must be **exactly** the same on both of the IPSec peers – they are compared during the ISAKMP Phase One authentication.

The following message is displayed – in the debug output – when preshared keys do not match:

```

%CRYPTO-6-IKMP_BAD_MESSAGE: IKE message from 10.1.1.1 failed its sanity check or is
malformed

```

Troubleshooting these problems consists of two steps:

- First, ISAKMP policies are compared to confirm both peers are using preshared keys authentication – **show crypto isakmp policy** command is used here;
- Second – the preshared keys should be compared on both peers to make sure both are **exactly** the same – the command **show crypto isakmp key** helps in this task:

```
show crypto isakmp key:
```

Keyring	Hostname/Address	Preshared Key
default	10.1.1.1	clsc0

- **Incorrect IPSec Access Lists** – as mentioned already Crypto ACLs are used to define what traffic should be encrypted and what traffic should be forwarded not encrypted.

Furthermore, ACLs on both IPSec peers should be symmetrical, i.e. the source IP address in the Crypto ACL on the first IPSec peer should be the destination IP address in the Crypto ACL on the second IPSec peer AND the destination IP address in the Crypto ACL on the first IPSec peer should be the source IP address in the Crypto ACL on the second IPSec peer – **show access-list** command helps to verify the Crypto ACLs;

- **Wrong Crypto Map placement** – applying the Crypto Map to the interface, makes this interface a termination point of IPSec tunnel – the Crypto Map should be applied to the **outbound** interface towards IPSec peer, so the traffic is encrypted and forwarded to the IPSec peer – **show crypto map interface interface\_name** verifies if the crypto map has been applied to the interface;
- **Routing issues** – as already mentioned, it is vital to make sure Layer 3 communications is established between IPSec peers before IPSec is configured – simple **ping** commands helps in confirming L3 connectivity.

The problems with routing include two scenarios:

- Problems with routing to the IPSec peer
- Problem with routing to the interface with Crypto Map applied

These issues has to be solved in order for IPSec services to work

## 4.2.Cisco PIX firewalls

Troubleshooting IPSec on Cisco PIX firewalls is similar to Cisco IOS routers – there is a set of display and information show commands:

- **Display show commands**

```
show crypto iskamp policy
```

```
Protection suite of priority 1
  encryption algorithm: Three key triple DES
  hash algorithm:       Secure Hash Standard
  authentication method: Pre-Shared Key
  Diffie-Hellman group: #1 (768 bit)
  lifetime:             86400 seconds, no volume limit
Default protection suite
  encryption algorithm: DES - Data Encryption Standard (56 bit keys).
  hash algorithm:       Secure Hash Standard
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group: #1 (768 bit)
```

lifetime: 86400 seconds, no volume limit

**show crypto ipsec transform-set**

```
Transform set TS: { esp-3des esp-sha-hmac }
will negotiate = { Tunnel, },
```

**show crypto map**

```
Crypto Map "CM" 10 ipsec-isakmp
Peer = 10.1.1.1
access-list IPsec-VPN; 1 elements
access-list IPsec-VPN line 1 permit ip 192.168.30.0 255.255.255.0 192.168.1.0
255.255.255.0 (hitcnt=46)
Current peer: 10.1.1.1
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ TS, }
```

• **Information show commands**

These commands allow to view the status of established IPsec-based VPN tunnels:

**show crypto engine**

```
Crypto Engine Connection Map:
size = 8, free = 6, used = 2, active = 2
```

**show crypto ipsec sa**

```
interface: outside
Crypto map tag: CM, local addr. 192.168.30.2

local ident (addr/mask/prot/port): (192.168.30.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
current_peer: 10.1.1.1:500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 130, #pkts encrypt: 130, #pkts digest 130
#pkts decaps: 112, #pkts decrypt: 112, #pkts verify 112
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress
failed: 0
#send errors 9, #recv errors 0

local crypto endpt.: 192.168.30.2, remote crypto endpt.: 10.1.1.1
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 488a59ce

inbound esp sas:
spi: 0x48d66fd(76375805)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 4, crypto map: CM
sa timing: remaining key lifetime (k/sec): (4607999/3514)
IV size: 8 bytes
replay detection support: Y

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0x488a59ce(1217026510)
```

```
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 3, crypto map: CM
sa timing: remaining key lifetime (k/sec): (4607999/3514)
IV size: 8 bytes
replay detection support: Y
```

outbound ah sas:

outbound pcp sas:

As with IPSec on Cisco IOS Routers, the usual reasons causing the problems with establishing the IPSec tunnels are listed below:

- **Incompatible ISAKMP policies** – **show crypto isakmp policy** command is used to verify the policies on both peers match along with **debug crypto isakmp** command;
- **Differing Preshared keys between the IPSec peers** – **show crypto isakmp** command verifies the method of authentication, which should be the same on both peers, i.e. **preshared**.

There is no way however to compare the keys on Cisco PIX boxes, as it is done on Cisco IOS Routers – once a preshared key is entered, there is no way to display it.

- **Incorrect IPSec Access Lists** – again the Crypto ACLs should be symmetrical on both ends of IPSec tunnel – **show access-list** helps in verifying Crypto ACLs;
- **Wrong Crypto Map placement** – **show crypto map** is a useful command to check what interface the Crypto Map is applied to;
- **Routing issues** – as mentioned already – Layer 3 connectivity is required before IPSec tunnels are established – **show route** command on PIX devices verifies the routing table and helps troubleshooting the problems with routing;

## 5. HEANET CONFERENCE – WORKSHOP EXERCISE

This section summarizes all the IPSec configuration aspects discussed so far using the workshop network as an example.

### 5.1. IPSec configuration on Cisco IOS routers

#### 5.1.1 Task 1) Preparation for IKE and IPSEC

The following information is required during the IKE and IPSec configuration:

- Key distribution method – **Manual**
- Authentication method – **Preshared Keys**
- IPSec Peer IP address – **10.1.1.1 255.255.255.0**
- IKE policy:
  - encryption – **3DES**
  - hash algorithm – **SHA1**
  - authentication method – **Preshared Keys** – use **clsc0** keyword
  - Diffie-Hellman group – **Group1**
- IPSec Policy
  - Transform set – **esp-3des esp-sha1-hmac**
  - Traffic to be protected – **all traffic from the local network to the remote network, i.e. 192.168.1.0 255.255.255.0**

#### 5.1.2 Task 2) IKE configuration

The commands below configure the IKE Phase One parameters:

```
crypto isakmp enable                # enables ISAKMP
crypto isakmp policy 1              # defines IKE policy
    encr 3des
    hash sha
    authentication pre-share
    group 1
    exit
crypto isakmp key clsc0 address 10.1.1.1 # defines the preshared key
```

#### 5.1.3 Task 3) IPSec configuration

The commands below help configuring the IPSec (IKE Phase Two) parameters

```
crypto ipsec transform-set TS esp-3des esp-sha1-hmac # defines transform set
```

Crypto ACLs are defined below – **please note** – use the ACL for your VLAN only:

```
# Crypto ACL for VLAN 10
```

## HEAnet Conference 2004 – Security Workshop

```
access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.1.0 0.0.0.255

# Crypto ACL for VLAN 20
access-list 101 permit ip 192.168.20.0 0.0.0.255 192.168.1.0 0.0.0.255

# Crypto ACL for VLAN 30 (PIX)
access-list 101 permit ip 192.168.30.0 255.255.255.0 192.168.1.0 255.255.255.0

# Crypto ACL for VLAN 40
access-list 101 permit ip 192.168.40.0 0.0.0.255 192.168.1.0 0.0.0.255

# Crypto ACL for VLAN 50
access-list 101 permit ip 192.168.50.0 0.0.0.255 192.168.1.0 0.0.0.255

crypto map CM 1 ipsec-isakmp          # defines crypto map
    set peer 10.1.1.1
    set transform-set TS
    match address 101
    exit

interface routers_outbound-interface # applies crypto map to the outbound interface
    crypto map CM
```

### 5.1.4 Task 4) Verifying IPSec

Below, the output of few **show** commands is included, helping in verification IPSec services configuration:

```
show crypto map isakmp policy
```

```
show crypto ipsec transform-set
```

```
show crypto ipsec sa
```

```
show crypto map
```

```
debug crypto isakmp & debug crypto ipsec:
```

Below is an output of successful IPSec tunnel creation process:

```
Kermit#sh debug
Cryptographic Subsystem:
  Crypto ISAKMP debugging is on
  Crypto IPSEC debugging is on
Kermit#
*Jan 11 02:42:01.513: ISAKMP (0:0): received packet from 10.1.1.3 dport 500 sport 500
Global (N) NEW SA
*Jan 11 02:42:01.513: ISAKMP: local port 500, remote port 500
```

## HEAnet Conference 2004 – Security Workshop

```
*Jan 11 02:42:01.513: ISAKMP: insert sa successfully sa = 63E23390
*Jan 11 02:42:01.513: ISAKMP (0:3): Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
*Jan 11 02:42:01.513: ISAKMP (0:3): Old State = IKE_READY New State = IKE_R_MM1

*Jan 11 02:42:01.513: ISAKMP (0:3): processing SA payload. message ID = 0
*Jan 11 02:42:01.517: ISAKMP: Looking for a matching key for 10.1.1.3 in default :
success
*Jan 11 02:42:01.517: ISAKMP (0:3): found peer pre-shared key matching 10.1.1.3
*Jan 11 02:42:01.517: ISAKMP (0:3) local preshared key found
*Jan 11 02:42:01.517: ISAKMP : Scanning profiles for xauth ...
*Jan 11 02:42:01.517: ISAKMP (0:3): Checking ISAKMP transform 1 against priority 1
policy
*Jan 11 02:42:01.517: ISAKMP:      encryption 3DES-CBC
*Jan 11 02:42:01.517: ISAKMP:      hash SHA
*Jan 11 02:42:01.517: ISAKMP:      default group 1
*Jan 11 02:42:01.517: ISAKMP:      auth pre-share
*Jan 11 02:42:01.517: ISAKMP:      life type in seconds
*Jan 11 02:42:01.517: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
*Jan 11 02:42:01.517: ISAKMP (0:3): atts are acceptable. Next payload is 0
! The IPsec peers found a matching ISAKMP policy

*Jan 11 02:42:01.525: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Jan 11 02:42:01.525: ISAKMP (0:3): Old State = IKE_R_MM1 New State = IKE_R_MM1

*Jan 11 02:42:01.525: ISAKMP (0:3): sending packet to 10.1.1.3 my_port 500 peer_port
500 (R) MM_SA_SETUP
*Jan 11 02:42:01.525: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE
*Jan 11 02:42:01.525: ISAKMP (0:3): Old State = IKE_R_MM1 New State = IKE_R_MM2

*Jan 11 02:42:01.633: ISAKMP (0:3): received packet from 10.1.1.3 dport 500 sport 500
Global (R) MM_SA_SETUP
*Jan 11 02:42:01.633: ISAKMP (0:3): Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
*Jan 11 02:42:01.633: ISAKMP (0:3): Old State = IKE_R_MM2 New State = IKE_R_MM3

*Jan 11 02:42:01.633: ISAKMP (0:3): processing KE payload. message ID = 0
*Jan 11 02:42:01.645: ISAKMP (0:3): processing NONCE payload. message ID = 0
*Jan 11 02:42:01.645: ISAKMP: Looking for a matching key for 10.1.1.3 in default :
success
*Jan 11 02:42:01.645: ISAKMP (0:3): found peer pre-shared key matching 10.1.1.3
*Jan 11 02:42:01.645: ISAKMP (0:3): SKEYID state generated
*Jan 11 02:42:01.645: ISAKMP (0:3): processing vendor id payload
*Jan 11 02:42:01.645: ISAKMP (0:3): speaking to another IOS box!
*Jan 11 02:42:01.645: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Jan 11 02:42:01.645: ISAKMP (0:3): Old State = IKE_R_MM3 New State = IKE_R_MM3

*Jan 11 02:42:01.645: ISAKMP (0:3): sending packet to 10.1.1.3 my_port 500 peer_port
500 (R) MM_KEY_EXCH
*Jan 11 02:42:01.645: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE
*Jan 11 02:42:01.645: ISAKMP (0:3): Old State = IKE_R_MM3 New State = IKE_R_MM4

*Jan 11 02:42:01.785: ISAKMP (0:3): received packet from 10.1.1.3 dport 500 sport 500
Global (R) MM_KEY_EXCH
*Jan 11 02:42:01.785: ISAKMP (0:3): Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
*Jan 11 02:42:01.785: ISAKMP (0:3): Old State = IKE_R_MM4 New State = IKE_R_MM5

*Jan 11 02:42:01.785: ISAKMP (0:3): processing ID payload. message ID = 0
*Jan 11 02:42:01.785: ISAKMP (0:3): ID payload
  next-payload : 8
  type          : 1
  address       : 10.1.1.3
  protocol      : 17
  port         : 500
  length       : 12
```

HEAnet Conference 2004 – Security Workshop

```
*Jan 11 02:42:01.785: ISAKMP (0:3): peer matches *none* of the profiles
*Jan 11 02:42:01.785: ISAKMP (0:3): processing HASH payload. message ID = 0
*Jan 11 02:42:01.785: ISAKMP (0:3): SA authentication status:      authenticated
*Jan 11 02:42:01.785: ISAKMP (0:3): SA has been authenticated with 10.1.1.3
! The peers are authenticated

*Jan 11 02:42:01.785: ISAKMP (0:3): peer matches *none* of the profiles
*Jan 11 02:42:01.785: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
*Jan 11 02:42:01.785: ISAKMP (0:3): Old State = IKE_R_MM5  New State = IKE_R_MM5

*Jan 11 02:42:01.785: ISAKMP (0:3): SA is doing pre-shared key authentication using id
type ID_IPV4_ADDR
*Jan 11 02:42:01.785: ISAKMP (0:3): ID payload
  next-payload : 8
  type          : 1
  address       : 10.1.1.1
  protocol      : 17
  port         : 500
  length       : 12
*Jan 11 02:42:01.785: ISAKMP (3): Total payload length: 12
*Jan 11 02:42:01.785: ISAKMP (0:3): sending packet to 10.1.1.3 my_port 500 peer_port
500 (R) MM_KEY_EXCH
*Jan 11 02:42:01.785: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE
*Jan 11 02:42:01.785: ISAKMP (0:3): Old State = IKE_R_MM5  New State = IKE_P1_COMPLETE

*Jan 11 02:42:01.785: ISAKMP (0:3): Input = IKE_MSG_INTERNAL, IKE_PHASE1_COMPLETE
*Jan 11 02:42:01.785: ISAKMP (0:3): Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

*Jan 11 02:42:01.797: ISAKMP (0:3): received packet from 10.1.1.3 dport 500 sport 500
Global (R) QM_IDLE
*Jan 11 02:42:01.797: ISAKMP: set new node -1070204650 to QM_IDLE
*Jan 11 02:42:01.797: ISAKMP (0:3): processing HASH payload. message ID = -1070204650
*Jan 11 02:42:01.797: ISAKMP (0:3): processing SA payload. message ID = -1070204650
*Jan 11 02:42:01.797: ISAKMP (0:3): Checking IPsec proposal 1
*Jan 11 02:42:01.797: ISAKMP: transform 1, ESP_3DES
*Jan 11 02:42:01.797: ISAKMP:   attributes in transform:
*Jan 11 02:42:01.797: ISAKMP:     encaps is 1 (Tunnel)
*Jan 11 02:42:01.797: ISAKMP:     SA life type in seconds
*Jan 11 02:42:01.797: ISAKMP:     SA life duration (basic) of 3600
*Jan 11 02:42:01.797: ISAKMP:     SA life type in kilobytes
*Jan 11 02:42:01.797: ISAKMP:     SA life duration (VPI) of  0x0 0x46 0x50 0x0

*Jan 11 02:42:01.797: ISAKMP:     authenticator is HMAC-SHA
*Jan 11 02:42:01.797: ISAKMP (0:3): atts are acceptable.
*Jan 11 02:42:01.797: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 10.1.1.1, remote= 10.1.1.3,
  local_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
  remote_proxy= 192.168.20.0/255.255.255.0/0/0 (type=4),
  protocol= ESP, transform= esp-3des esp-sha-hmac (Tunnel),
  lifedur= 0s and 0kb,
  spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
*Jan 11 02:42:01.797: IPSEC(kei_proxy): head = CM, map->ivrf = , kei->ivrf =
*Jan 11 02:42:01.797: ISAKMP (0:3): processing NONCE payload. message ID = -1070204650
*Jan 11 02:42:01.797: ISAKMP (0:3): processing ID payload. message ID = -1070204650
*Jan 11 02:42:01.797: ISAKMP (0:3): processing ID payload. message ID = -1070204650
! Matching IPsec policy has been negotiated and authenticated, Next SAs are set up

*Jan 11 02:42:01.797: ISAKMP (0:3): asking for 1 spis from ipsec
*Jan 11 02:42:01.797: ISAKMP (0:3): Node -1070204650, Input = IKE_MSG_FROM_PEER,
IKE_QM_EXCH
*Jan 11 02:42:01.797: ISAKMP (0:3): Old State = IKE_QM_READY  New State =
IKE_QM_SPI_STARVE
*Jan 11 02:42:01.797: IPSEC(key_engine): got a queue event...
*Jan 11 02:42:01.797: IPSEC(spi_response): getting spi 3068806418 for SA
  from 10.1.1.1      to 10.1.1.3      for prot 3
*Jan 11 02:42:01.797: ISAKMP: received ke message (2/1)
*Jan 11 02:42:02.049: ISAKMP (0:3): Creating IPsec SAs
*Jan 11 02:42:02.049:      inbound SA from 10.1.1.3 to 10.1.1.1 (f/i)  0/ 0
```

## HEAnet Conference 2004 – Security Workshop

```
(proxy 192.168.20.0 to 192.168.1.0)
*Jan 11 02:42:02.049:      has spi 0xB6EA4512 and conn_id 2000 and flags 2
*Jan 11 02:42:02.049:      lifetime of 3600 seconds
*Jan 11 02:42:02.049:      lifetime of 4608000 kilobytes
*Jan 11 02:42:02.049:      has client flags 0x0
*Jan 11 02:42:02.049:      outbound SA from 10.1.1.1      to 10.1.1.3
      (f/i) 0/ 0 (proxy 192.168.1.0      to 192.168.20.0      )
*Jan 11 02:42:02.049:      has spi -1315901426 and conn_id 2001 and flags A
*Jan 11 02:42:02.049:      lifetime of 3600 seconds
*Jan 11 02:42:02.049:      lifetime of 4608000 kilobytes
*Jan 11 02:42:02.049:      has client flags 0x0
*Jan 11 02:42:02.049: ISAKMP (0:3): sending packet to 10.1.1.3 my_port 500 peer_port
500 (R) QM_IDLE
*Jan 11 02:42:02.049: ISAKMP (0:3): Node -1070204650, Input = IKE_MSG_FROM_IPSEC,
IKE_SPI_REPLY
*Jan 11 02:42:02.049: ISAKMP (0:3): Old State = IKE_QM_SPI_STARVE New State =
IKE_QM_R_QM2
*Jan 11 02:42:02.049: IPSEC(key_engine): got a queue event...
*Jan 11 02:42:02.049: IPSEC(initialize_sas): ,
      (key eng. msg.) INBOUND local= 10.1.1.1, remote= 10.1.1.3,
      local_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
      remote_proxy= 192.168.20.0/255.255.255.0/0/0 (type=4),
      protocol= ESP, transform= esp-3des esp-sha-hmac (Tunnel),
      lifedur= 3600s and 4608000kb,
      spi= 0xB6EA4512(3068806418), conn_id= 2000, keysize= 0, flags= 0x2
*Jan 11 02:42:02.049: IPSEC(initialize_sas): ,
      (key eng. msg.) OUTBOUND local= 10.1.1.1, remote= 10.1.1.3,
      local_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
      remote_proxy= 192.168.20.0/255.255.255.0/0/0 (type=4),
      protocol= ESP, transform= esp-3des esp-sha-hmac (Tunnel),
      lifedur= 3600s and 4608000kb,
      spi= 0xB190F00E(2979065870), conn_id= 2001, keysize= 0, flags= 0xA
*Jan 11 02:42:02.049: IPSEC(kei_proxy): head = CM, map->ivrf = , kei->ivrf =
*Jan 11 02:42:02.049: IPSEC(crypto_ipsec_sa_find_ident_head): reconnecting with the
same proxies and 10.1.1.3
*Jan 11 02:42:02.049: IPSEC(add mtree): src 192.168.1.0, dest 192.168.20.0, dest_port
0

*Jan 11 02:42:02.049: IPSEC(create_sa): sa created,
      (sa) sa_dest= 10.1.1.1, sa_prot= 50,
      sa_spi= 0xB6EA4512(3068806418),
      sa_trans= esp-3des esp-sha-hmac , sa_conn_id= 2000
*Jan 11 02:42:02.049: IPSEC(create_sa): sa created,
      (sa) sa_dest= 10.1.1.3, sa_prot= 50,
      sa_spi= 0xB190F00E(2979065870),
      sa_trans= esp-3des esp-sha-hmac , sa_conn_id= 2001
```

**! IPSec SAs are set up and data can be securely exchanged**

## 5.2. IPSec configuration on Cisco PIX Firewalls

### 5.2.1 Task 1) Preparation for IKE and IPSEC

The following information is required during the IKE and IPsec configuration:

- Key distribution method – **Manual**
- Authentication method – **Preshared Keys**
- IPsec Peer IP address – **10.1.1.1 255.255.255.0**
- IKE policy:
  - encryption – **3DES**
  - hash algorithm – **SHA1**
  - authentication method – **Preshared Keys** – use **clsc0** keyword
  - Diffie-Hellman group – **Group1**
- IPsec Policy
  - Transform set – **esp-3des esp-sha1-hmac**
  - Traffic to be protected – **all traffic from the local network to the remote network, i.e. 192.168.1.0 255.255.255.0**

### 5.2.2 Task 2) IKE configuration

```
isakmp enable outside # enables ISAKMP

# defines the policy and its priority
iskamp policy 1 encryption 3des # defines encryption
iskamp policy 1 hash sha # defines the hash algorithm
iskamp policy 1 authentication pre-share # defines how peers authenticate
iskamp policy 1 group 1 # defines Diffie-Hellman algorithm group

isakmp key clsc0 address 10.1.1.1 # defines the pre-shared key
```

### 5.2.3 Task 3) IPsec configuration

The commands below help configuring the IPsec (IKE Phase Two) parameters:

First, transform set is defined:

```
crypto ipsec transform set TS esp-3des esp-sha-hmac # defines transform set
```

Then, Crypto ACLs are defined:

```
# Crypto ACL for VLAN 30 (PIX)
access-list 101 permit ip 192.168.30.0 255.255.255.0 192.168.1.0 255.255.255.0
```

Next Crypto Map is created and applied to the outside interface:

```
crypto map CM 1 ipsec-isakmp
crypto map CM 1 set transform-set TS
crypto map CM 1 match address 101
crypto map CM 1 set peer 10.1.1.1

crypto map CM interface outside      # applies crypto map to the outbound interface
```

#### 5.2.4 Task 4) Verifying IPsec

Below, the output of few **show** commands is included, helping in verification IPsec services configuration:

```
show crypto map isakmp policy

show crypto ipsec transform-set

show crypto ipsec sa

show crypto map

debug crypto isakmp
```

## 6. MEMORY AND CPU CONSIDERATIONS

Packets that are processed by IPSec will be slower than packets not processed through IPSec. There are several reasons for this and they may cause significant performance problems:

- IPSec introduces packet expansion, which is more likely to require fragmentation and the corresponding reassembly of IPSec datagrams;
- Encrypted packets will probably be authenticated, which means that there are two cryptographic operations being performed for every packet
- The authentication algorithms are slow (although work has been done to speed up things as the Diffie-Hellman computations).

In addition, the Diffie-Hellman key exchange used in IKE is an exponentiation of very large numbers (between 768 and 1024 bytes) and can take up to four seconds on a Cisco 2500. Performance of RSA is dependent on the size of the prime number chosen for the RSA key pair.

For each router, the SA database will take up approximately 300 bytes, plus 120 bytes for every SA therein. In situations where there are two IPSec SAs, one inbound and one outbound, 540 bytes are required; this would be in most cases. Each IKE SA entry is approximately 64 bytes each. The only time when one IPSec SA for a dataflow is needed is when one-way communication is chosen.

IPSec and IKE will impact performance when active. Diffie-Hellman key exchanges, public key authentication, and encryption/decryption will consume a significant amount of resources, although great lengths have been gone through to minimize this impact.

There should be a small decrease in performance for non-encrypted packets going through an interface that is doing crypto because all packets have to be checked against the crypto map. There should be no performance impact on packets traversing the router that avoid an interface doing crypto. The biggest impact will be on the encrypted data flows themselves.

To minimize the impact of the crypto subsystem on the rest of the router, it is recommended to use Group 1 for Diffie-Hellman key exchanges within IKE, MD5 as the hashing algorithm, and longer lifetimes. The trade-off for this performance tuning is weaker cryptography. Ultimately, it is up to the customer's security policy to determine which features to use and which to leave alone.

## 7. GLOSSARY

**Authentication Header (AH):** A security protocol that provides authentication and optional replay-detection services. AH is embedded in the data to be protected (a full IP datagram, for example). AH can be used either by itself or with Encryption Service Payload (ESP). Refer to the RFC 2402;

**Authentication:** One of the functions of the IPSec framework. Authentication establishes the integrity of datastream and ensures that it is not tampered with in transit. It also provides confirmation about datastream origin;

**Certification Authority (CA):** A third-party entity that is responsible for issuing and revoking certificates. Each device that has its own certificate and public key of the CA can authenticate every other device within a given CA's domain. This term is also applied to server software that provides these services;

**Certificate:** A cryptographically signed object that contains an identity and a public key associated with this identity;

**Classic crypto:** Cisco proprietary encryption mechanism used in Cisco IOS® Software Release 11.2;

**Certificate Revocation List (CRL):** A digitally signed message that lists all of the current but revoked certificates listed by a given CA. This is analogous to a book of stolen charge card numbers that allow stores to reject bad credit cards;

**Crypto map:** A Cisco IOS software configuration entity that performs two primary functions: (1) it selects data flows that need security processing and (2) defines the policy for these flows and the crypto peer that traffic needs to go to;

A crypto map is applied to an interface. The concept of a crypto map was introduced in classic crypto but was expanded for IPSec.

**Data integrity:** Data integrity mechanisms, through the use of secret-key based or public-key based algorithms, that allow the recipient of a piece of protected data to verify that the data has not been modified in transit;

**Data confidentiality:** Method where protected data is manipulated so that no attacker can read it. This is commonly provided by data encryption and keys that are only available to the parties involved in the communication;

**Data origin authentication:** A security service where the receiver can verify that protected data could have originated only from the sender. This service requires a data integrity service plus a key distribution mechanism, where a secret key is shared only between the sender and receiver;

**Data Encryption Standard (DES):** The DES was published in 1977 by the National Bureau of Standards 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 crypto (40-bit and 56-bit key lengths), IPSec crypto (56-bit key), and on the PIX Firewall (56-bit key).

**Diffie-Hellman:** A method of establishing a shared key over an insecure medium.

Diffie-Hellman is a component of Oakley (see definition below);

**DSS:** A digital signature algorithm designed by The US National Institute of Standards and Technology (NIST) based on public key cryptography. DSS does not do user datagram encryption. DSS is a component in classic crypto, as well as the Redcreek IPsec card, but not in IPsec implemented in Cisco IOS software;

**Encryption Service Adapter (ESA):** A hardware based encryption accelerator that is used in:

- Cisco 7204 and 7206 routers;
- Second-generation Versatile Interface Processor2-40s (VIP2-40s) in all Cisco 7500 series routers
- VIP2-40 in the Cisco 7000 series routers that have the Cisco 7000 series Route Switch Processor (RSP7000) and Cisco 7000 series Chassis Interface (RSP7000CI) cards installed.

IPsec does not use the ESA acceleration, but will work in a box that has an ESA card on a software-only basis.

**Encapsulating Security Payload (ESP):** Security protocol that provides data confidentiality and protection with optional authentication and replay-detection services - ESP completely encapsulates user data. ESP can be used either by itself or in conjunction with AH. Check out RFC 2406: IP Encapsulating Security Payload (ESP);

**Hash:** A one way function that takes an input message of arbitrary length and produces a fixed length digest. Cisco uses both Secure Hash Algorithm (SHA) and Message Digest 5 (MD5) hashes within its implementation of the IPsec framework (see HMAC below).

**HMAC:** A mechanism for message authentication using cryptographic hashes such as SHA and MD5. For an exhaustive discussion of HMAC, check out RFC 2104;

**Internet Key Exchange (IKE):** 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/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 CA service, or the secure DNS (DNSSec). This 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, as can be seen in the next definition.

**Internet Security Association and Key Management Protocol (ISAKMP):** A protocol framework that defines the mechanics of implementing a key exchange protocol and negotiation of a security policy. ISAKMP is defined in the Internet Security Association and Key Management Protocol (ISAKMP).

**IPsec NAT Transparency:** The IPsec NAT Transparency feature introduces support for IP Security (IPsec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known

incompatibilities between NAT and IPSec. NAT Traversal is a feature that is auto detected by VPN devices. There are no configuration steps for a router running Cisco IOS Software Release 12.2(13)T above. If both VPN devices are NAT-T capable, NAT Traversal is auto detected and auto negotiated.

**ISAKMP/Oakley:** See IKE.

**Message Digest 5 (MD5):** A one way hashing algorithm that produces a 128-bit hash. Both MD5 and Secure Hash Algorithm (SHA) are variations on MD4, which is designed to strengthen the security of this hashing algorithm. SHA is more secure than MD4 and MD5. Cisco uses hashes for authentication within the IPSec framework.

**Oakley:** A key exchange protocol that defines how to acquire authenticated keying material. The basic mechanism for Oakley is the Diffie-Hellman key exchange algorithm. The standard is described in RFC 2412: The OAKLEY Key Determination Protocol;

**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 were to break a key, PFS ensures that the attacker would not be able to derive any other key. If PFS were not enabled, someone could hypothetically 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 SAs setup by this IKE SA. With PFS, breaking IKE would not give an attacker immediate access to IPSec. The attacker would have to break each IPSec SA individually. Cisco's IOS IPSec implementation uses PFS group 1 (D-H 768 bit) by default.

**Replay-detection:** A security service where the receiver can reject old or duplicate packets in order to defeat replay attacks (replay attacks rely on the attacker sending out older or duplicate packets to the receiver and the receiver thinking that the bogus traffic is legitimate). Replay-detection is done by using sequence numbers combined with authentication, and is a standard feature of IPSec.

**RSA:** A public key cryptographic algorithm (named after its inventors, Rivest, Shamir and Adleman) with a variable key length. RSA's main weakness is that it is significantly slow to compute compared to popular secret-key algorithms, such as DES. Cisco's IKE implementation uses a Diffie-Hellman exchange to get the secret keys. This exchange can be authenticated with RSA (or pre-shared keys). With the Diffie-Hellman 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 public domain, and must be licensed from RSA Data Security.

**Security Association (SA):** An instance of security policy and keying material applied to a data flow. Both IKE and IPSec use SAs, although SAs are independent of one another. IPSec SAs are unidirectional and they are unique in each security protocol. A set of SAs is needed for a protected data pipe, one per direction per protocol. For example, if there is a pipe that supports ESP between peers, one ESP SA is required for each direction. SAs are uniquely identified by destination (IPSec endpoint) address, security protocol (AH or ESP), and security parameter index (SPI).

IKE negotiates and establishes SAs on behalf of IPSec. A user can also establish IPSec SAs manually.

An IKE SA is used by IKE only, and unlike the IPSec SA, it is bi-directional.

**Secure Hash Algorithm (SHA):** A one way hash put forth by 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-SHA for authentication.

**Transport Mode:** An encapsulation mode for AH/ESP. Transport Mode encapsulates the upper layer payload (such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)) of the original IP datagram. This mode can only be used when the peers are the endpoints of the communication. The contrast of Transport Mode is Tunnel Mode.

**Tunnel Mode:** 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 Virtual Private Network (VPN) scenario).

## 8.CISCO ROUTERS AND PIX FIREWALLS WORKSHOP CONFIGURATION OUTPUTS

### 8.1.Basic configuration – no IPSec

#### 8.1.1Central Router

```
version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Kermit
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
ip cef
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex half
!
!
interface FastEthernet3/0
 ip address 10.1.1.1 255.255.255.0
 duplex half
!
ip classless
ip route 192.168.10.0 255.255.255.0 10.1.1.2
ip route 192.168.20.0 255.255.255.0 10.1.1.3
ip route 192.168.30.0 255.255.255.0 10.1.1.4
ip route 192.168.40.0 255.255.255.0 10.1.1.5
ip route 192.168.50.0 255.255.255.0 10.1.1.6
no ip http server
no ip http secure-server
!
!
gatekeeper
 shutdown
!
!
line con 0
 transport preferred all
 transport output all
 stopbits 1
line aux 0
 transport preferred all
 transport output all
 stopbits 1
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
!
end
```

### 8.1.2 Branch router

The configuration of branch routers is exactly the same – only interfaces' IP addresses differ – therefore configuration output from one router is included here only:

```

version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot system flash c3725-ik9s-mz.123-10a.bin
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
ip cef
!
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0
 no ip address
 shutdown
 clockrate 2000000
 no fair-queue
!
interface FastEthernet0/1
 ip address 10.1.1.2 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
 clockrate 2000000
!
ip http server
no ip http secure-server
ip classless
ip route 192.168.1.0 255.255.255.0 10.1.1.1
!
!
line con 0
 transport preferred all
 transport output all
line aux 0
 transport preferred all
 transport output all
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
end

```

**8.1.3 Cisco PIX firewall**

```

PIX Version 6.3(4)
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto shutdown
interface ethernet3 auto shutdown
interface ethernet4 auto shutdown
interface ethernet5 auto shutdown
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security4
nameif ethernet3 intf3 security6
nameif ethernet4 intf4 security8
nameif ethernet5 intf5 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
names

# This allows basic connectivity check - ping
access-list 100 permit icmp any any

pager lines 24
mtu outside 1500
mtu inside 1500
mtu intf2 1500
mtu intf3 1500
mtu intf4 1500
mtu intf5 1500
ip address outside 192.168.30.2 255.255.255.0
ip address inside 192.168.60.1 255.255.255.0
no ip address intf2
no ip address intf3
no ip address intf4
no ip address intf5
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
no failover ip address outside
no failover ip address inside
no failover ip address intf2
no failover ip address intf3
no failover ip address intf4
no failover ip address intf5
pdm history enable
arp timeout 14400
static (inside,outside) 192.168.30.3 192.168.60.2 netmask 255.255.255.255 0 0
access-group 100 in interface outside
route outside 0.0.0.0 0.0.0.0 192.168.30.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute

```

HEAnet Conference 2004 – Security Workshop

```
aaa-server TACACS+ protocol tacacs+
aaa-server TACACS+ max-failed-attempts 3
aaa-server TACACS+ deadtime 10
aaa-server RADIUS protocol radius
aaa-server RADIUS max-failed-attempts 3
aaa-server RADIUS deadtime 10
aaa-server LOCAL protocol local
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
Cryptochecksum:27aed242f6334aec90f9a26bebeae98
: end
```

## 8.2.IPsec configuration

### 8.2.1Central Router

```

version 12.3
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Kermit
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
ip cef
!
!
crypto isakmp policy 1
  encr 3des
  authentication pre-share
crypto isakmp key clsc0 address 10.1.1.2
crypto isakmp key clsc0 address 10.1.1.3
crypto isakmp key clsc0 address 10.1.1.4
crypto isakmp key clsc0 address 10.1.1.5
crypto isakmp key clsc0 address 10.1.1.6
!
!
crypto ipsec transform-set TS esp-3des esp-sha-hmac
!
crypto map CM 1 ipsec-isakmp
  set peer 10.1.1.2
  set peer 10.1.1.3
  set peer 10.1.1.4
  set peer 10.1.1.5
  set peer 10.1.1.6
  set transform-set TS
  match address 101
!
!
interface FastEthernet0/0
  ip address 192.168.1.1 255.255.255.0
  duplex half
!
!
interface FastEthernet3/0
  ip address 10.1.1.1 255.255.255.0
  duplex half
  crypto map CM
!
ip classless
ip route 192.168.10.0 255.255.255.0 10.1.1.2
ip route 192.168.20.0 255.255.255.0 10.1.1.3
ip route 192.168.30.0 255.255.255.0 10.1.1.4
ip route 192.168.40.0 255.255.255.0 10.1.1.5
ip route 192.168.50.0 255.255.255.0 10.1.1.6
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.10.0 0.0.0.255
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.20.0 0.0.0.255
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.30.0 0.0.0.255

```

## HEAnet Conference 2004 – Security Workshop

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.40.0 0.0.0.255
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.50.0 0.0.0.255
!
!
gatekeeper
 shutdown
!
!
line con 0
 transport preferred all
 transport output all
 stopbits 1
line aux 0
 transport preferred all
 transport output all
 stopbits 1
line vty 0 4
 login
 transport preferred all
 transport input all
 transport output all
!
!
end
```

### 8.2.2 Branch router

```
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot system flash c3725-ik9s-mz.123-10a.bin
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
ip cef
!
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
crypto isakmp key c1sc0 address 10.1.1.1
!
!
crypto ipsec transform-set TS esp-3des esp-sha-hmac
!
crypto map CM 1 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set TS
 match address 101
!
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0
 no ip address
 shutdown
 clockrate 2000000
```

```

no fair-queue
!
interface FastEthernet0/1
ip address 10.1.1.2 255.255.255.0
duplex auto
speed auto
crypto map CM
!
interface Serial0/1
no ip address
shutdown
clockrate 2000000
!
ip http server
no ip http secure-server
ip classless
ip route 192.168.1.0 255.255.255.0 10.1.1.1
!
!
access-list 101 permit ip 192.168.10.0 0.0.0.255 192.168.1.0 0.0.0.255
!
!
line con 0
transport preferred all
transport output all
line aux 0
transport preferred all
transport output all
line vty 0 4
login
transport preferred all
transport input all
transport output all
!
end

```

### 8.2.3 Cisco PIX firewall

```

PIX Version 6.3(4)
interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto shutdown
interface ethernet3 auto shutdown
interface ethernet4 auto shutdown
interface ethernet5 auto shutdown
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security4
nameif ethernet3 intf3 security6
nameif ethernet4 intf4 security8
nameif ethernet5 intf5 security10
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
names

# This allows basic connectivity check - ping

```

## HEAnet Conference 2004 – Security Workshop

```
access-list 100 permit icmp any any

access-list IPSec-VPN permit ip 192.168.30.0 255.255.255.0 192.168.1.0 255.255.255.0
pager lines 24
mtu outside 1500
mtu inside 1500
mtu intf2 1500
mtu intf3 1500
mtu intf4 1500
mtu intf5 1500
ip address outside 192.168.30.2 255.255.255.0
ip address inside 192.168.60.1 255.255.255.0
no ip address intf2
no ip address intf3
no ip address intf4
no ip address intf5
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
no failover ip address outside
no failover ip address inside
no failover ip address intf2
no failover ip address intf3
no failover ip address intf4
no failover ip address intf5
pdm history enable
arp timeout 14400
static (inside,outside) 192.168.30.3 192.168.60.2 netmask 255.255.255.255 0 0
access-group 100 in interface outside
route outside 0.0.0.0 0.0.0.0 192.168.30.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server TACACS+ max-failed-attempts 3
aaa-server TACACS+ deadtime 10
aaa-server RADIUS protocol radius
aaa-server RADIUS max-failed-attempts 3
aaa-server RADIUS deadtime 10
aaa-server LOCAL protocol local
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
crypto ipsec transform-set TS esp-3des esp-sha-hmac
crypto map CM 10 ipsec-isakmp
crypto map CM 10 match address IPSec-VPN
crypto map CM 10 set peer 10.1.1.1
crypto map CM 10 set transform-set TS
crypto map CM interface outside
isakmp enable outside
isakmp key ***** address 10.1.1.1 netmask 255.255.255.255
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption 3des
isakmp policy 1 hash sha
isakmp policy 1 group 1
isakmp policy 1 lifetime 86400
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
Cryptochecksum:27aed242f6334aec90f9a26bebeae98
: end
```

**9.NETWORK DIAGRAMS**

Included are network diagrams from HEAnet Conference Workshop.