IPv6 IPsec Availability

Merike Kaeo

merike@doubleshotsecurity.com

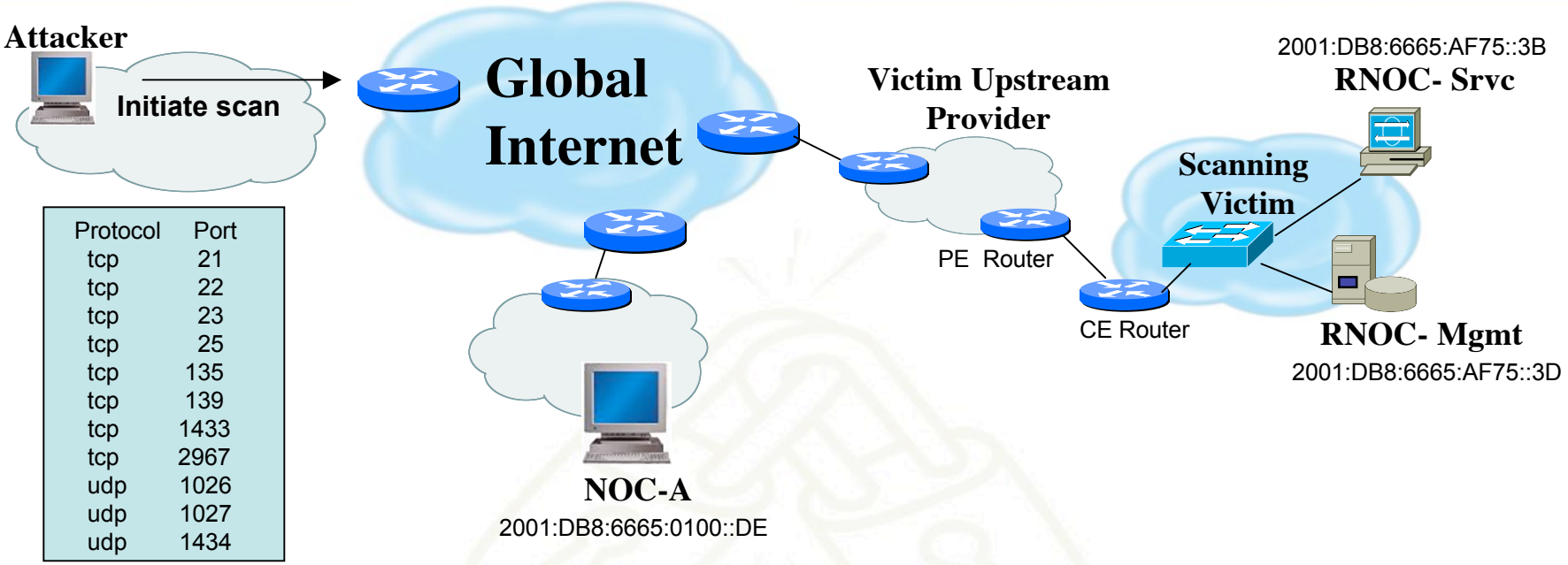
Why Use IPsec ?

- Confidentiality….although not the only reason…….
- Data integrity and source authentication
  - Data “signed” by sender and “signature” verified by the recipient
  - Modification of data can be detected by signature “verification”
  - Because “signature” based on a shared secret, it gives source authentication
    - The shared secret is cryptographically derived
- Anti-replay protection
  - Optional : the sender must provide it but the recipient may ignore
- Key Management
  - IKE – session negotiation and establishment
  - Sessions are rekeyed or deleted automatically
  - Secret keys are securely established and authenticated
  - Remote peer is authenticated through varying options

Considerations For Using IPsec

- Security Services
  - Data origin authentication
  - Data integrity
  - Replay protection
  - Confidentiality
- Size of network
- How trusted are end hosts
- Vendor support
- What other mechanisms can accomplish similar attack risk mitigation

Protecting Against Scanning Attacks



Protocol	Port
tcp	21
tcp	22
tcp	23
tcp	25
tcp	135
tcp	139
tcp	1433
tcp	2967
udp	1026
udp	1027
udp	1434

IPsec Security Policy Database

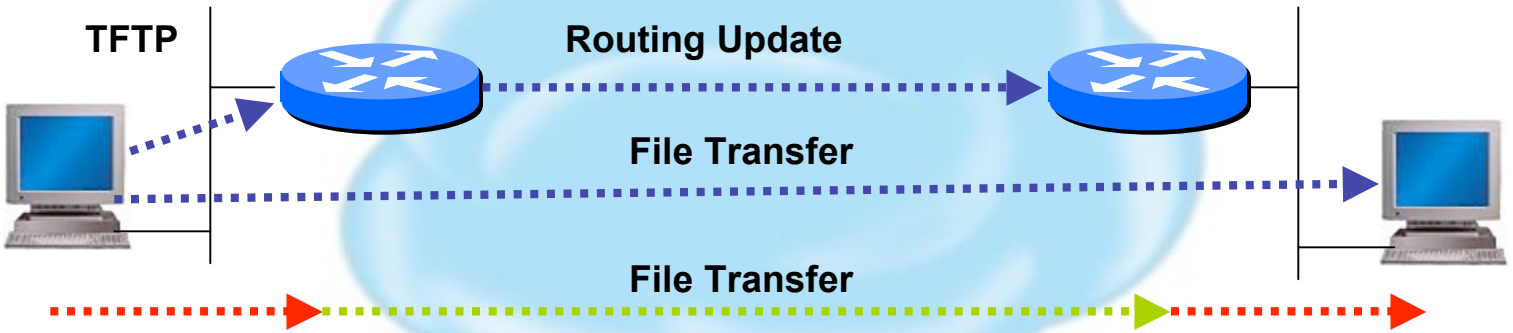
From	To	Protocol	Dst Port	Policy
2001:DB8:6665:0100::DE	2001:DB8:6665:01C8::3B	TCP / UDP	53 (DNS)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3B	TCP	25 (SNMP)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3D	UDP	1812/1813 (RADIUS)	ESP: SHA1, AES-128
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3D	UDP	514 (Syslog)	ESP: SHA1, 3DES
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::/48	TCP / UDP	ANY	ESP: SHA1

IPv6 Architectures using IPsec

- Protect all traffic using IPsec for data origin authentication and integrity
  - AH versus ESP/Null Encryption
- Add confidentiality as dictated by security policy
  - ESP

Need to dispel myth that using IPsec mandates the demise of network layer defense mechanisms

Transport vs Tunnel Mode

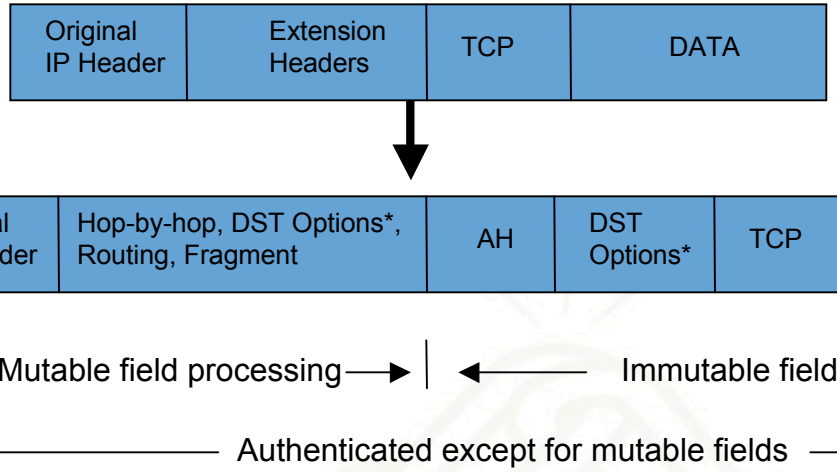


Transport Mode: End systems are the initiator and recipient of protected traffic

Tunnel Mode: Gateways act on behalf of hosts to protect traffic

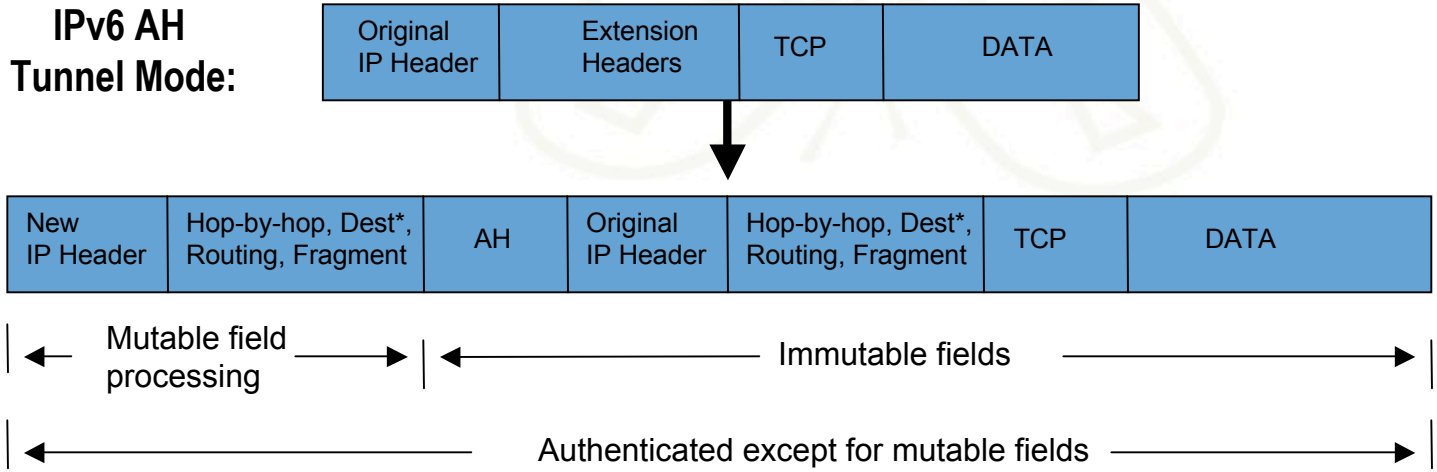
IPv6 IPsec AH

IPv6 AH Transport Mode:



Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

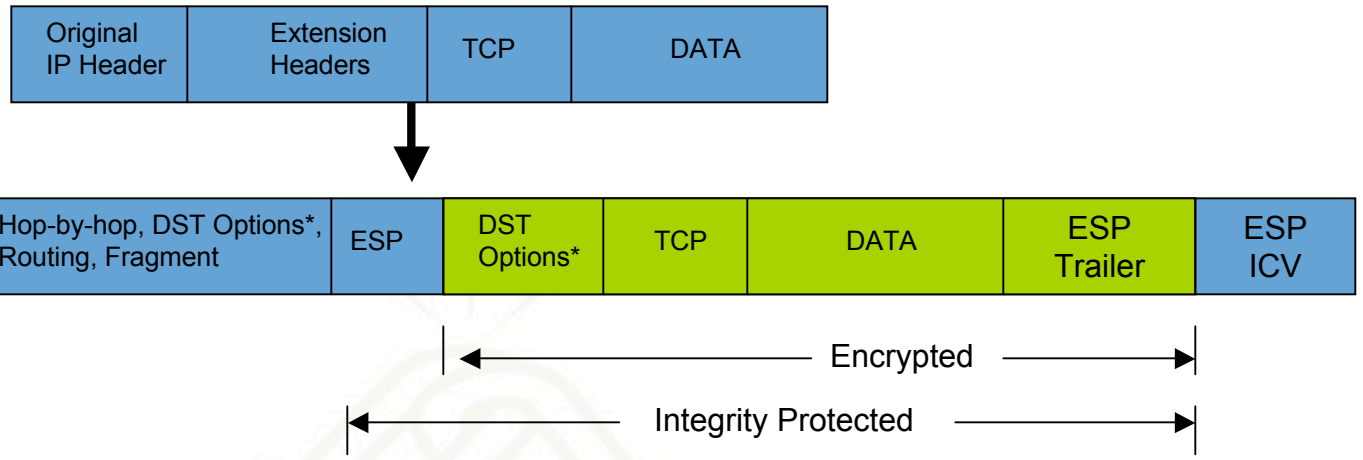
IPv6 AH Tunnel Mode:



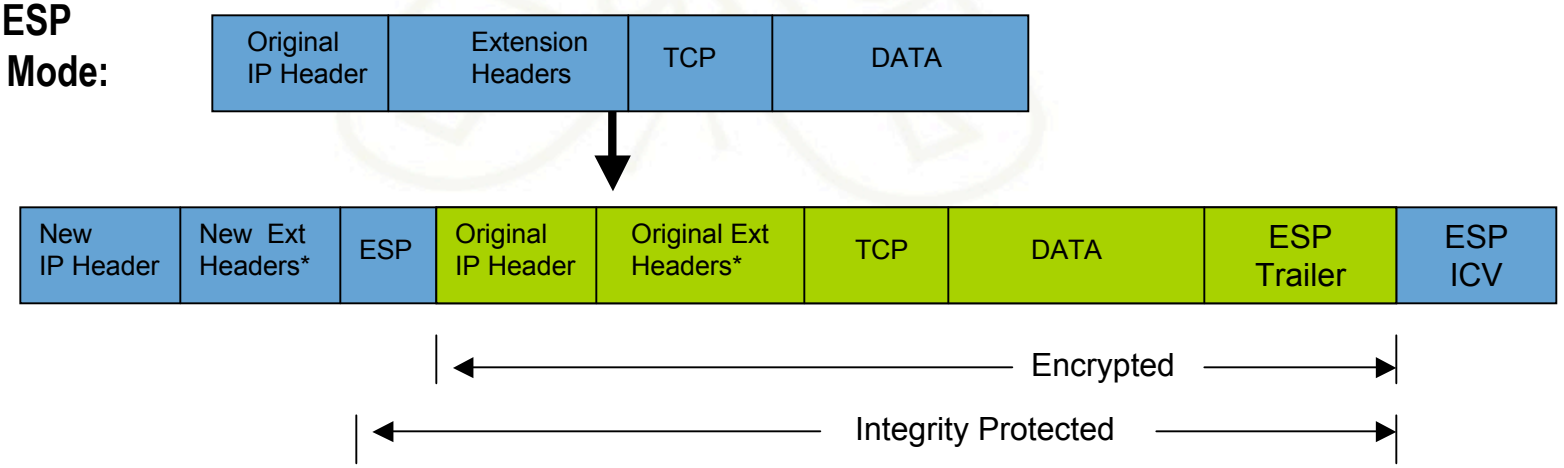
Mutable Fields:
- DSCP
- ECN
- Flow Label
- Hop Limit

IPv6 IPsec ESP

IPv6 ESP Transport Mode:



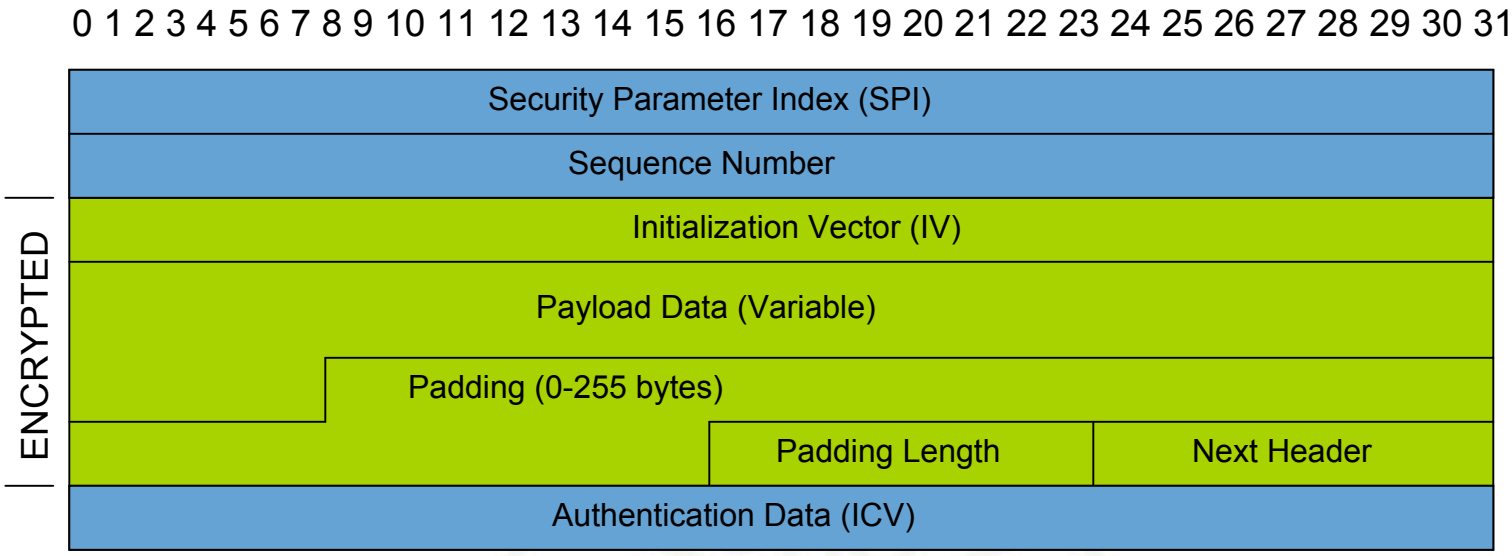
IPv6 ESP Tunnel Mode:



Enhancements Needed

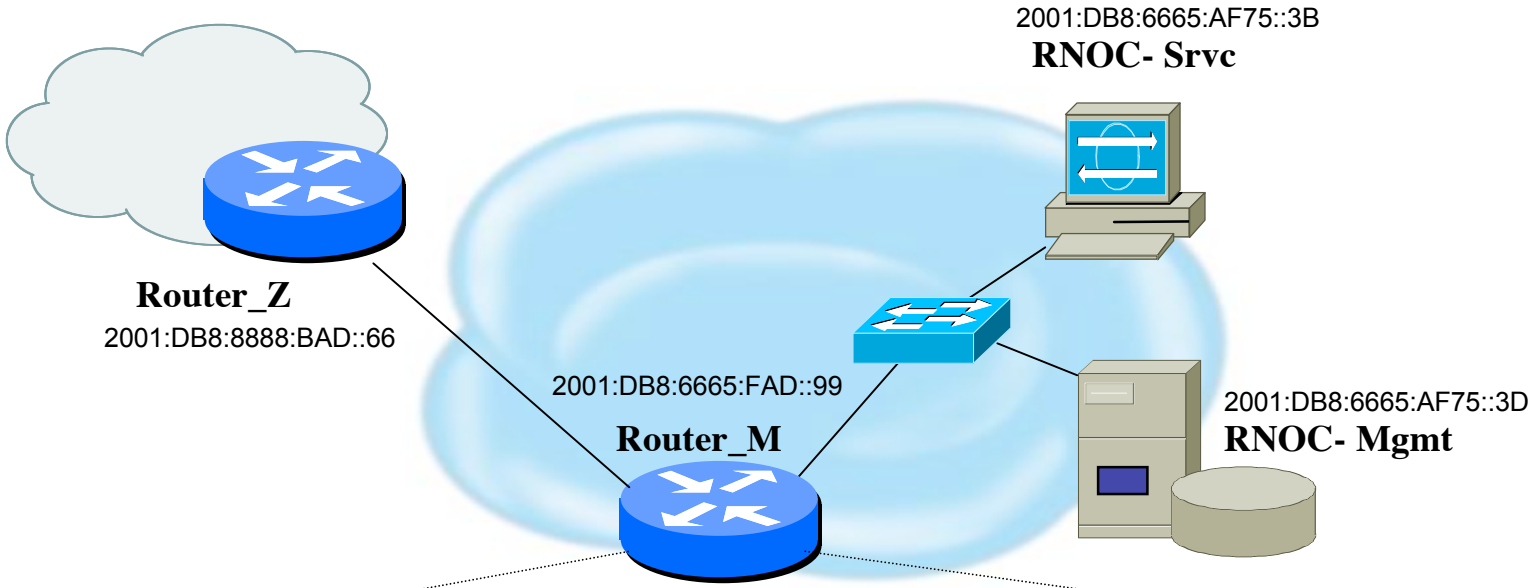
- Standards Modifications
  - Need to take into consideration Stateless Autoconfiguration where Router Advertisement sends network prefix
  - Need to be able to differentiate between encrypted versus integrity protected traffic
- Usability
  - Interoperable defaults
  - Consistent terminology

ESP Header Format



SPI: Arbitrary 32-bit number that specifies SA to the receiving device
Seq #: Start at 1 and must never repeat; receiver may choose to ignore
IV: Used to initialize CBC mode of an encryption algorithm
Payload Data: Encrypted IP header, TCP or UDP header and data
Padding: Used for encryption algorithms which operate in CBC mode
Padding Length: Number of bytes added to the data stream (may be 0)
Next Header: The type of protocol from the original header which appears in the encrypted part of the packet
Auth Data: ICV is a digital signature over the packet and it varies in length depending on the algorithm used (SHA-1, MD5)

Potentially Easy Configuration



Syslog server 2001:DB8:6665:AF75::3D authenticate esp-null sha1 pre-share ‘secret4syslog’

TFTP server 2001:DB8:6665:AF75::3D authenticate esp-null aes128 pre-share ‘secret4tftp’

BGP peer 2001:DB8:8888:BAD::66 authenticate esp-null aes128 pre-share ‘secret4AS#XXX’

Interoperable Defaults For SAs

- Security Association groups elements of a conversation together
  - AH authentication algorithm and keys
  - ESP encryption algorithm and key(s)
  - Cryptographic synchronization
  - SA lifetime
  - SA source address
  - Mode (transport or tunnel)

How Do We Communicate Securely ?

Do we want integrity protection of data ?

Do we want to keep data confidential ?

Which algorithms do we use ?

What are the key lengths ?

When do we want to create new keys ?

Are we providing security end-to-end ?

IKE - Internet Key Exchange

- Automatically establishes SAs and creates/deletes cryptographic material
- Authenticates communicating peers
- Works in 2 Phases
  - Phase I
    - Establish a secure channel (ISAKMP/IKE SA) to negotiate the data protection cryptographic material
    - Results in a single ISAKMP/IKE SA
  - Phase II
    - Establishes the secure channel for the transmission of data
    - Results in a pair of IPsec SAs

IKEv1

- Phase I
  - Establish a secure communication channel (ISAKMP/IKE SA)
  - Main Mode
    - Negotiates an ISAKMP SA which will be used to create IPsec SAs
    - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DF group to use)
    - Do a Diffie-Hellman exchange
    - Provide authentication information
    - Authenticate the peer
  - Aggressive Mode
    - Uses 3 (vs 6) messages to establish IKE SA
    - No denial of service protection
    - Does not have identity protection
    - Optional exchange and not widely implemented
- Phase II
  - Establishes a secure channel for actual data (IPsec SA)
  - Quick mode
    - All traffic is encrypted using the ISAKMP/IKE Security Association
    - Each quick mode negotiation results in two IPsec Security Associations
    - Creates/refreshes keys

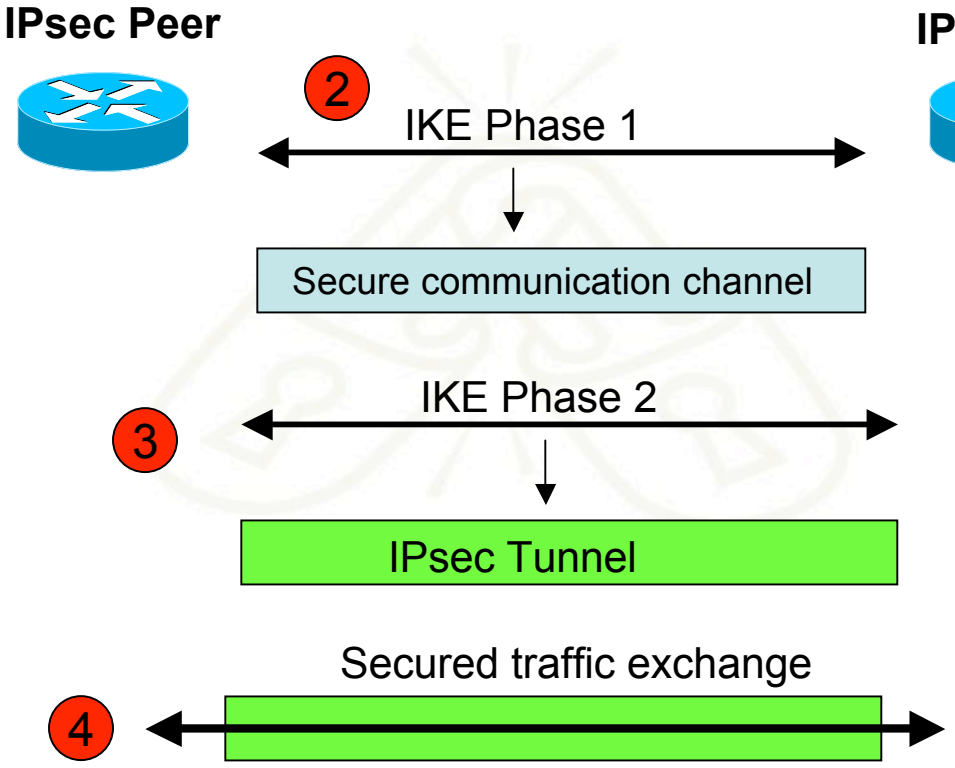
IPsec with IKE

Peers Authenticate using:

- Pre-shared key
- Digital Certificate

1

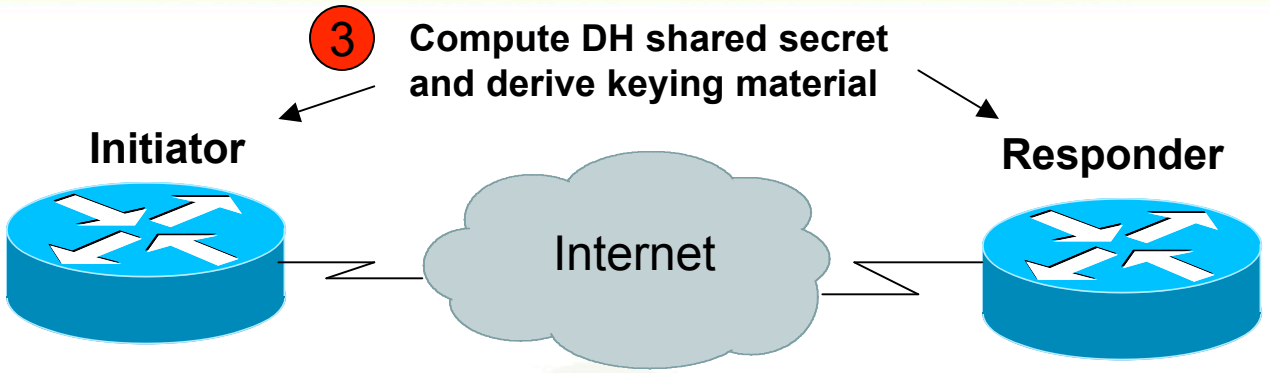
Traffic which needs to be protected is recognized as requiring IPsec protection



IPsec IKE Phase 1 Uses DH Exchange

- First public key algorithm (1976)
- Diffie Hellman is a key establishment algorithm
  - Two parties in a DF exchange can generate a shared secret
  - There can even be N-party DF changes where N peers can all establish the same secret key
- Diffie Hellman can be done over an insecure channel
- IKE authenticates a Diffie-Hellman exchange
  - Pre-shared secret
  - Nonce (RSA signature)
  - Digital signature

IKE Phase 1 Main Mode



1 Negotiate IKE Policy

IKE Message 1 (SA proposal) →

← IKE Message 2 (accepted SA)

2 Authenticated DH Exchange

IKE Message 3 (DH public value, nonce) →

← IKE Message 4 (DH public value, nonce)

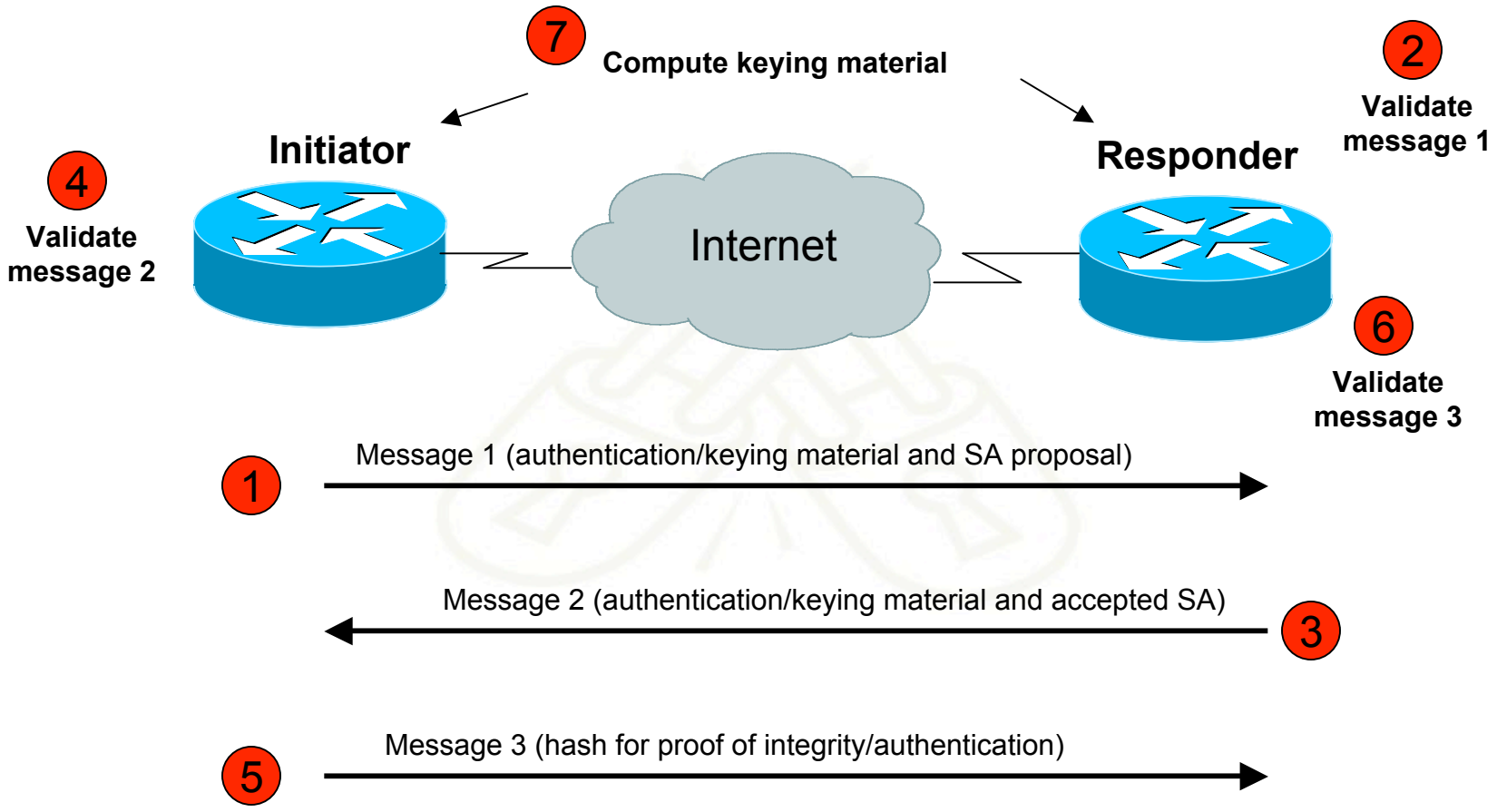
4 Protect IKE Peer Identity

IKE Message 5 (Authentication material, ID) →

← IKE Message 6 (Authentication material, ID)

(Encrypted)

IKE Phase 2 Quick Mode



PFS- what is it?

- Perfect Forward Secrecy
- Doing new DH exchange to derive keying material (instead of using the shared key derived from previous phase 1 and not doing another phase 1)
- Only relevant for re-keying a phase 2 IKE (I.e. automatically establishing new keys for integrity and confidentiality of the traffic you want to protect)

(DH used to derive shared secret which is used to derive keying material for IPsec security services)

Pretty Good IPsec Policy

- IKE Phase 1 (aka ISAKMP SA or IKE SA or Main Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (480 min = 28800 sec)
  - SHA-1
  - DH Group 14 (aka MODP# 14)
- IKE Phase 2 (aka IPsec SA or Quick Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (60 min = 3600 sec)
  - SHA-1
  - PFS 2
  - DH Group 14 (aka MODP# 14)

Vendor Specific Deployment Issues

- Lack of interoperable defaults
  - A default does NOT mandate a specific security policy
  - Defaults can be modified by end users
- Configuration complexity
  - Too many knobs
  - Vendor-specific terminology
- Good News: IPv6 support in most current implementations

Routers: Configuring IPsec

- For IPv6, consider using transport mode between routers and syslog servers, tftp servers, snmp servers, etc.
- Document for Cisco IPv6 IPsec configuration:
  - http://www.lseltd.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/v6_ipsec.pdf
- Document for Juniper IPsec configuration:
  - http://www.pacificbroadband.com/techpubs/software/junos/junos83/feature-guide-83/html/fg-ipsec13.html#1139838

Cisco: Configuring IPsec

STEP 1 *Configure the IKE Phase 1 Policy (ISAKMP Policy)*

Cisco literature refers to IKE Phase 1 as the ISAKMP policy.
It is configured using the command:

crypto isakmp policy *priority*

Multiple policies can be configured and the priority number, which ranges from 1 to 10,000, denotes the order of preference that a given policy will be negotiated with an ISAKMP peer. The lower value has the higher priority. Once in the ISAKMP configuration mode, the following parameters can be specified are:

- Encryption Algorithm
- Hash Algorithm
- Authentication Method
- Group Lifetime

Cisco: Configuring IPsec

STEP 2 *Set the ISAKMP Identity*

The ISAKMP identity specifies how the IKE Phase 1 peer is identified, which can be either by IP address or host name.
The command to use is:

crypto isakmp *identity* {*IP address* | *hostname*}

By default, a peer's ISAKMP identity is the peer's IP address.
If you decide to change the default just keep in mind that it is best to always be consistent across your entire IPsec-protected network in the way you choose to define a peer's identity.

Cisco: Configuring IPsec

STEP 3 *Configure the IPsec AH and ESP Parameters*

The AH and ESP parameters are configured with the following commands:

crypto ipsec transform-set *transform-set-name* <transform 1> <transform 2>
mode [tunnel | transport]
crypto ipsec security-association lifetime seconds *seconds*

STEP 4 *Configure the IPsec Traffic Selectors*

The traffic selectors are configured by defining extended access-lists.
The *permit* keyword causes all IP traffic that matches the specified conditions to be protected by IPsec

Cisco: Configuring IPsec

STEP 5 *Configure the IKE Phase 2 (IPsec SA) Policy*

This step sets up a crypto map which specifies all the necessary parameters to negotiate the IPsec SA policy. The following commands are required:

crypto map *crypto-map-name seq-num* ipsec-isakmp
match address *access-list-id*
set peer [*IP address* | *hostname*]
set transform-set *transform-set-name*
set security-association lifetime seconds *seconds*
set pfs [group1 | group 2]

Cisco: Configuring IPsec

STEP 6 *Apply the IPsec Policy to an Interface*

The configured crypto map is then applied to the appropriate interface using the crypto map *crypto-map-name* command. It is possible to apply the same crypto map to multiple interfaces. This case would require the use of the command:

crypto map *crypto-map-name* local-address *interface-id*

Using this command, the identifying interface will be used as the local address for IPsec traffic originating from or destined to those interfaces sharing the same crypto map. A loopback interface should be used as the identifying interface.

Unix IPsec IKE Daemons

- Racoon2 (IKEv1 and IKEv2 and KINK)
  - http://www.racoon2.wide.ad.jp/w/
- Ipsec-tools (IKEv1)
  - port of KAME's IPsec utilities to the Linux-2.6 IPsec implementation; it supports NetBSD and FreeBSD as well
  - http://ipsec-tools.sourceforge.net/
- Strongswan (IKEv1 and IKEv2)
  - http://www.strongswan.org/
- Openikev2 (IKEv2)
  - http://openikev2.sourceforge.net/

LINUX and MACOSX machines

- Type command ‘ *man racoon* ’
  - Read how to set-up racoon, the name for this particular IKE software
- Type command ‘ *man setkey* ’
  - This command is used to set up the SA database
- The following files are located in ***/etc/raccoon***:
  - ***psk.txt*** – file which contains the shared secrets
  - ***raccoon.conf*** – file which configures IKE phase 1 and IKE phase 2 parameters

Set Up Security Policy Database

- Create a file named ' ***ipsec.conf*** ' which will be used with *setkey* to establish the correct security associations. The file should have the following information:

  - `flush;`
  - `spdflush;`
  - `spdadd 2001:DB8:6665:AF75::3D/128 2001:DB8:8888:BAD::66/128 any -P out ipsec esp/transport//require ;`
  - `spdadd 2001:DB8:8888:BAD::66/128 2001:DB8:6665:AF75::3D/128 any -P in ipsec esp/transport//require ;`

Creating SA Database

- Test to see what happens when you try and create an SA database:
- Type the following:
  - `setkey -f /etc/racoon/ipsec.conf`
- Use the '`setkey -P -D`' command to see if appropriate entries have been created

Pre-Shared Key Configuration

- Edit the psk.txt file to add the peer IP address and the pre-shared secret key:
  - `# file for pre-shared keys used for IKE authentication`
  - `# format is:  'identifier' 'key'`
  - `# For example:`
  - `#  10.1.1.1  flibbertigibbet`
  - `#  www.example.com  12345`
  - `#  foo@www.example.com  micropachycephalosaurus`
  - `<peer IPv6 address>  <shared secret>`
- Since the psk.txt file contains sensitive information make sure that the file is appropriately protected:
  - `chmod 600 /etc/raccoon/psk.txt`

Racoon.conf file

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format
      and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
log debug;
remote anonymous
{
    {
        exchange_mode main;
        lifetime time 480 min;
        proposal {
            encryption_algorithm 3des;
            hash_algorithm sha1;
            authentication_method pre_shared_key;
            dh_group 14;
        }
    }
```

```
sainfo anonymous
{
    pfs_group 2;
    lifetime time 60 min ;
    encryption_algorithm 3des,
    blowfish 448, rijndael ;
    authentication_algorithm
    hmac_sha1, hmac_md5 ;
    compression_algorithm deflate ;
}
```

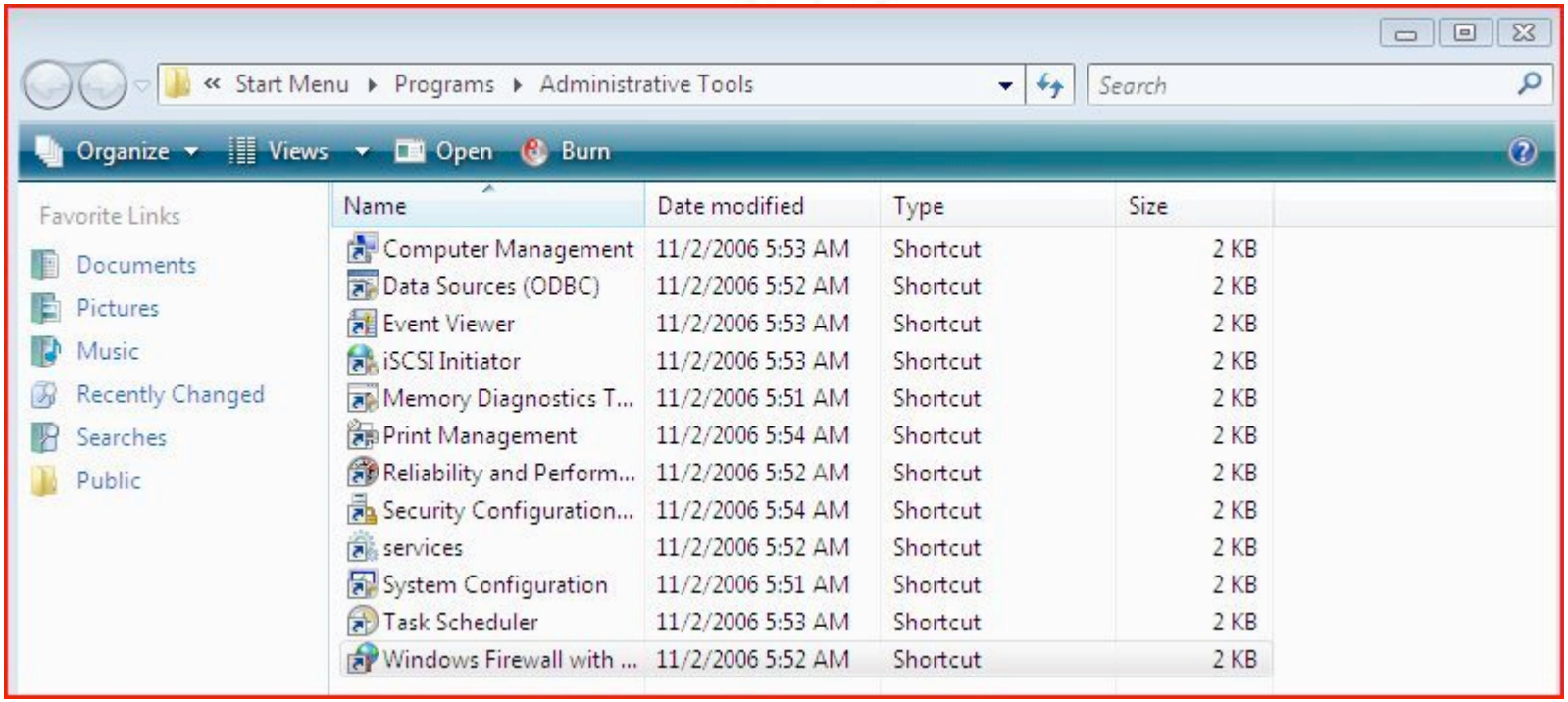
Testing Racoon

- Test racoon with the following command:
  - `racoon -v -f /etc/racoon/racoon.conf -l /etc/racoon/test.log`

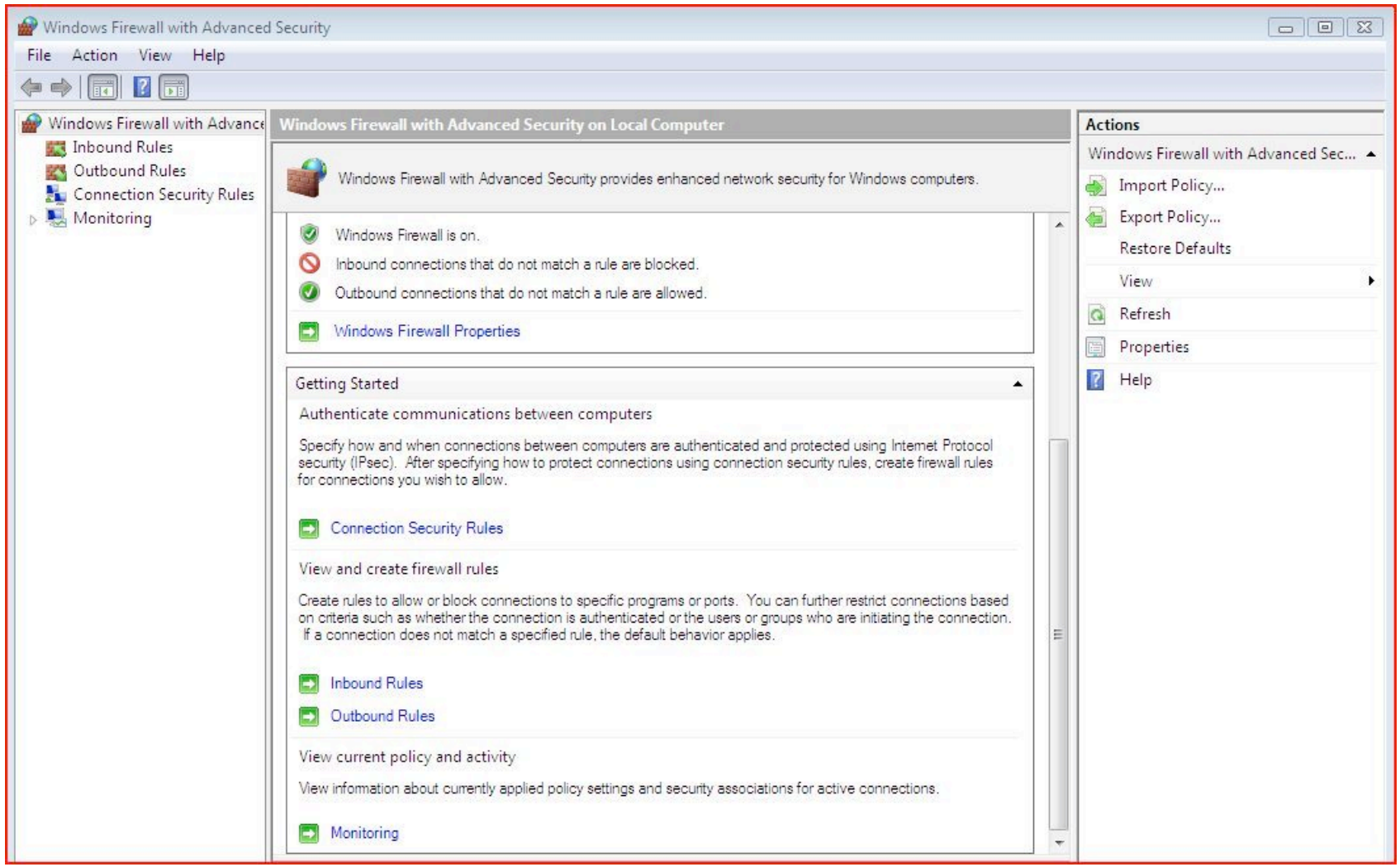
- The '`-l /etc/racoon/test.log`' file is used to write any debug information in the event that there are problems.

Vista: Configuring IPsec

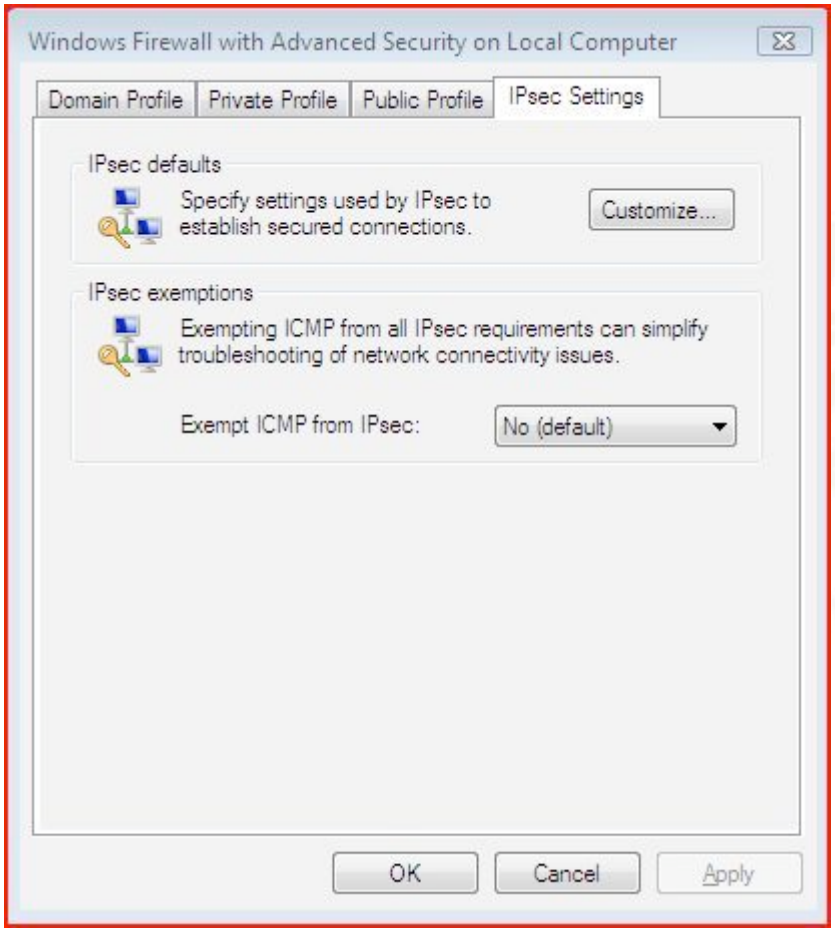
- Defaults work great in a MS-only environment
- Need to edit firewall (wf.mmc) otherwise

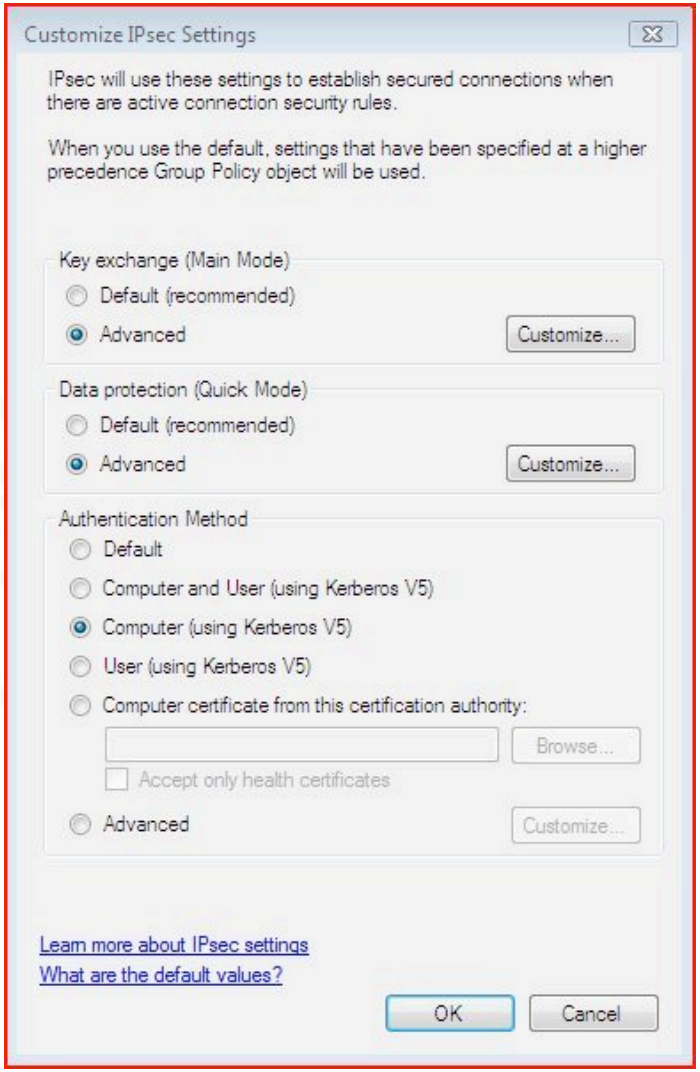


Vista: Configuring IPsec

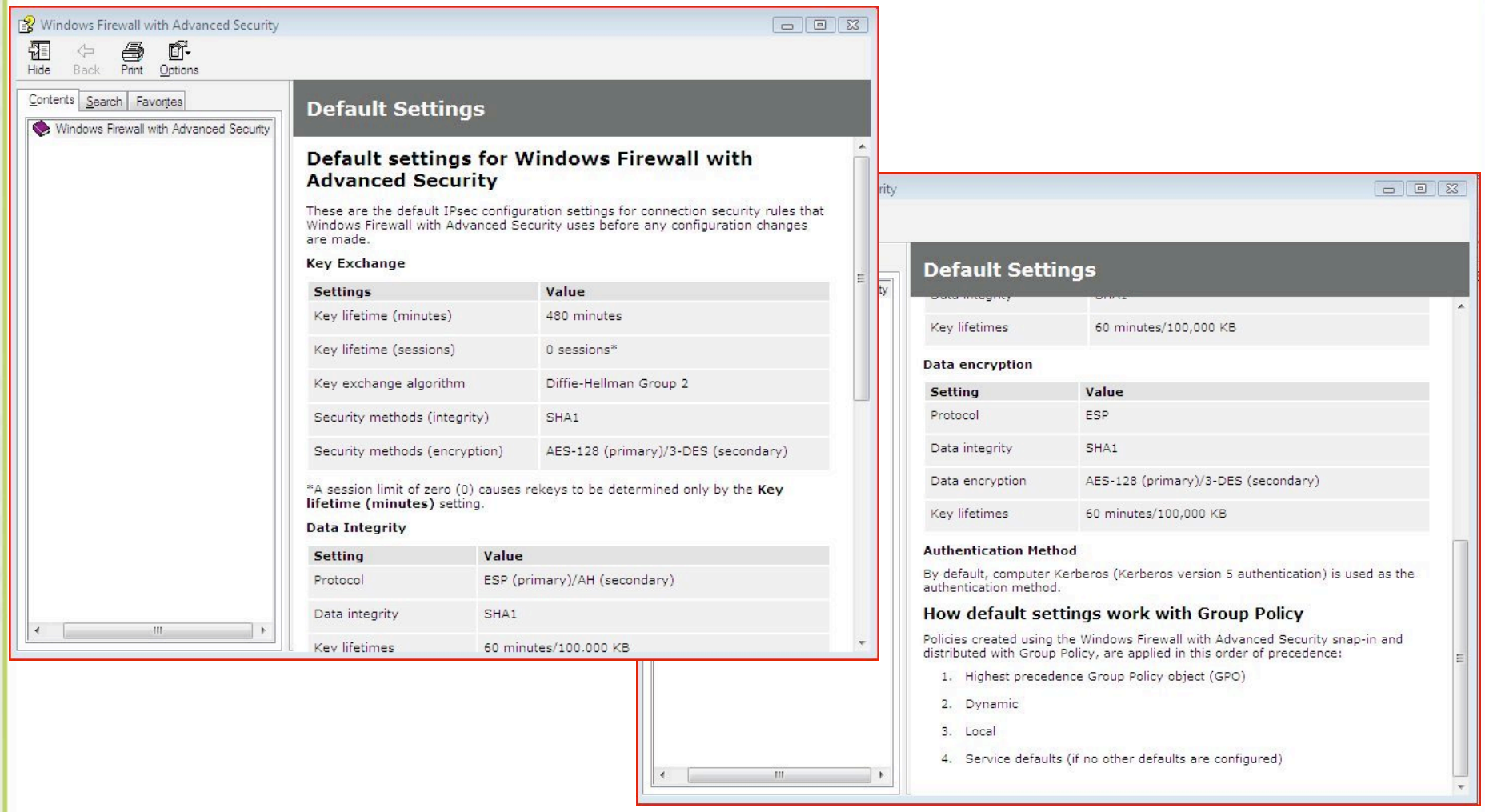


Vista: Customizing IPsec Settings

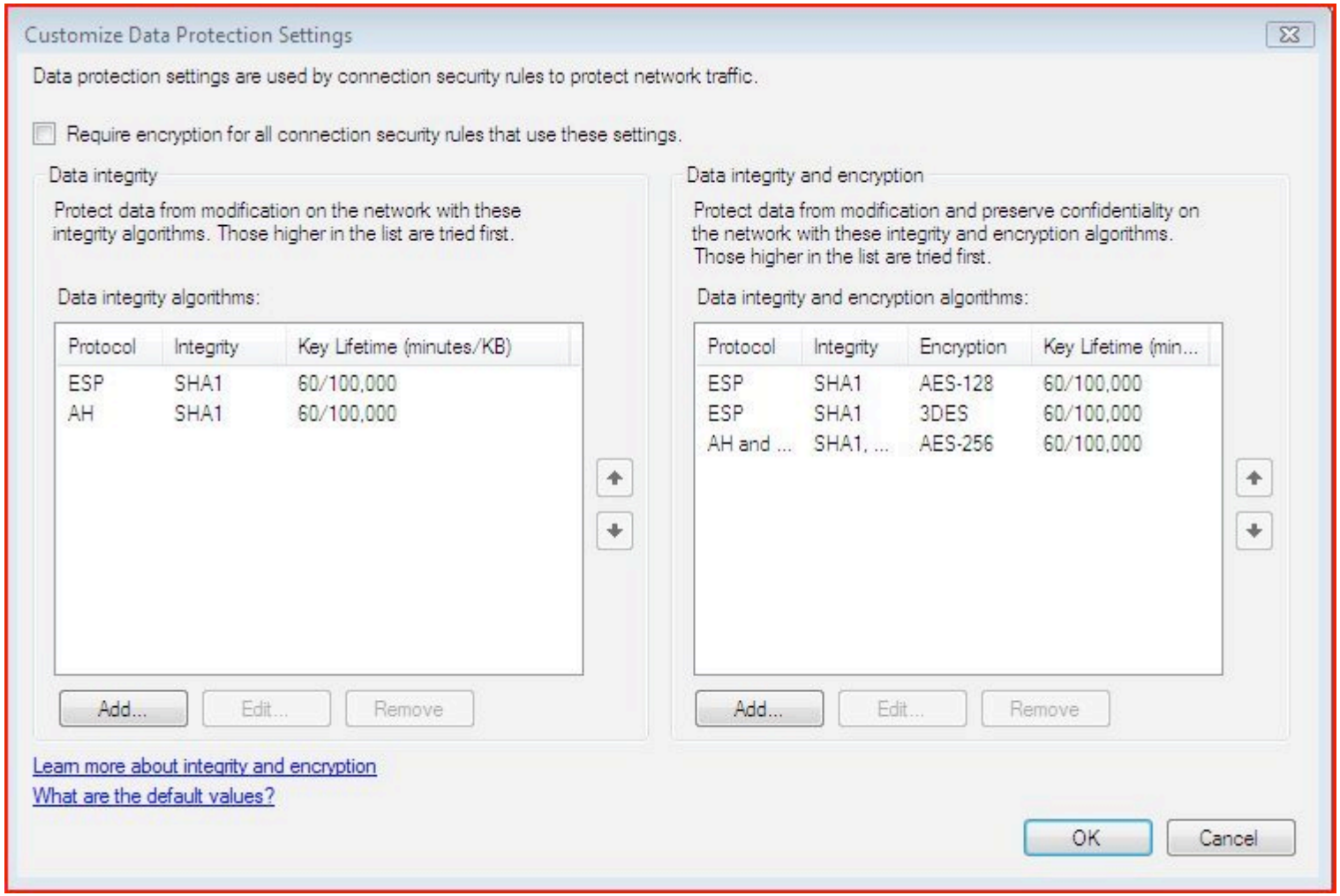




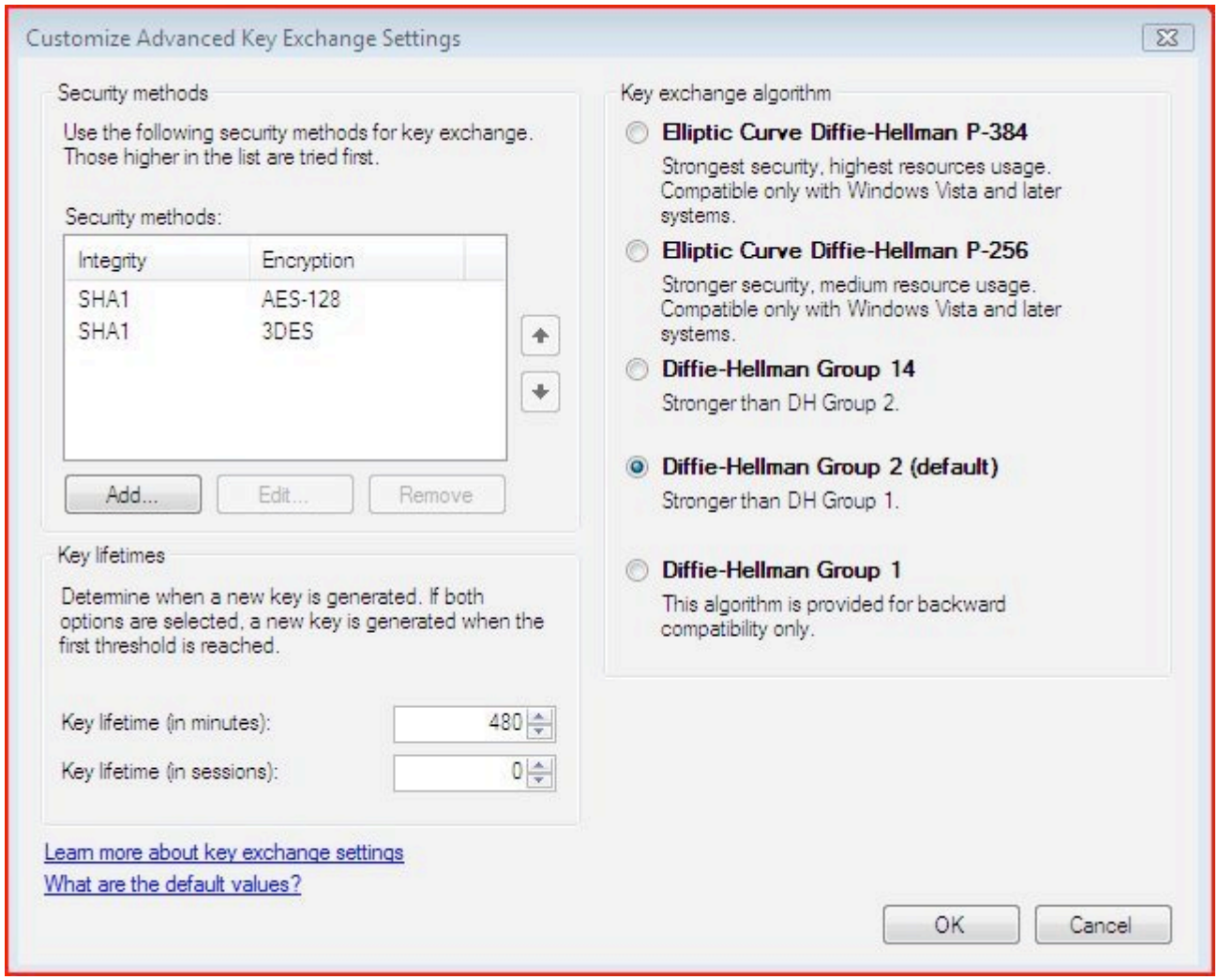
Vista IPsec Defaults



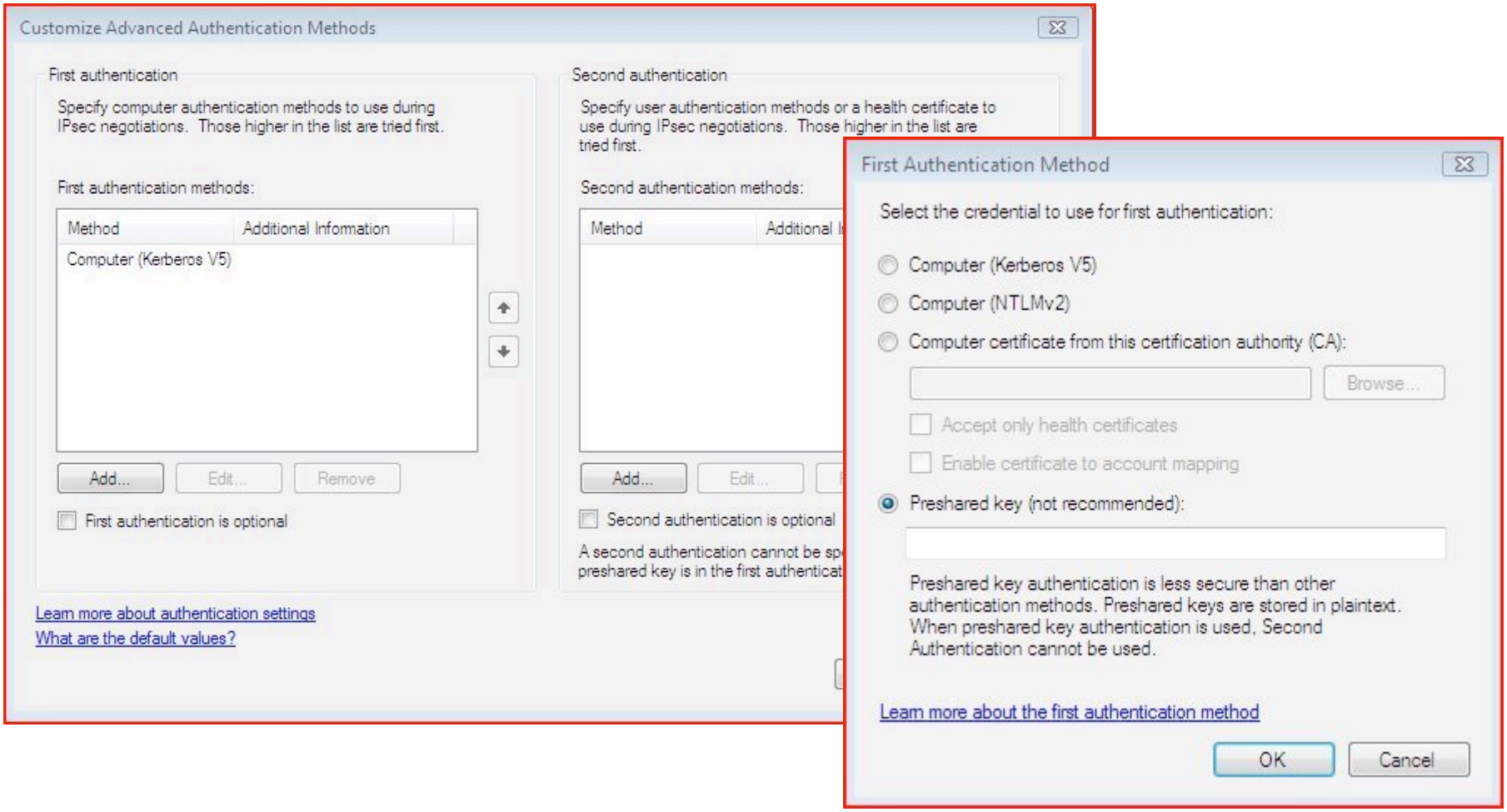
Vista: Customizing Data Protection



Vista: Customizing Key Exchange



Vista: Customizing Authentication



Conclusions

- IPsec is a complex standard but user configurations shouldn't be
- Using IPsec does NOT mean you have to encrypt the data (providing traffic integrity can be useful too)
- Don't leave IPsec out when you are trying to gain experience with IPv6 - time to fix usability issues is NOW