



# IPv6 IPsec Availability

Merike Kaeo

[merike@doubleshotsecurity.com](mailto: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



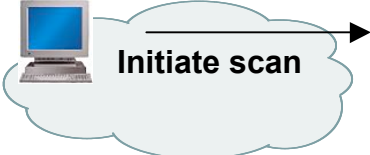
# Considerations For Using IPsec

- **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**

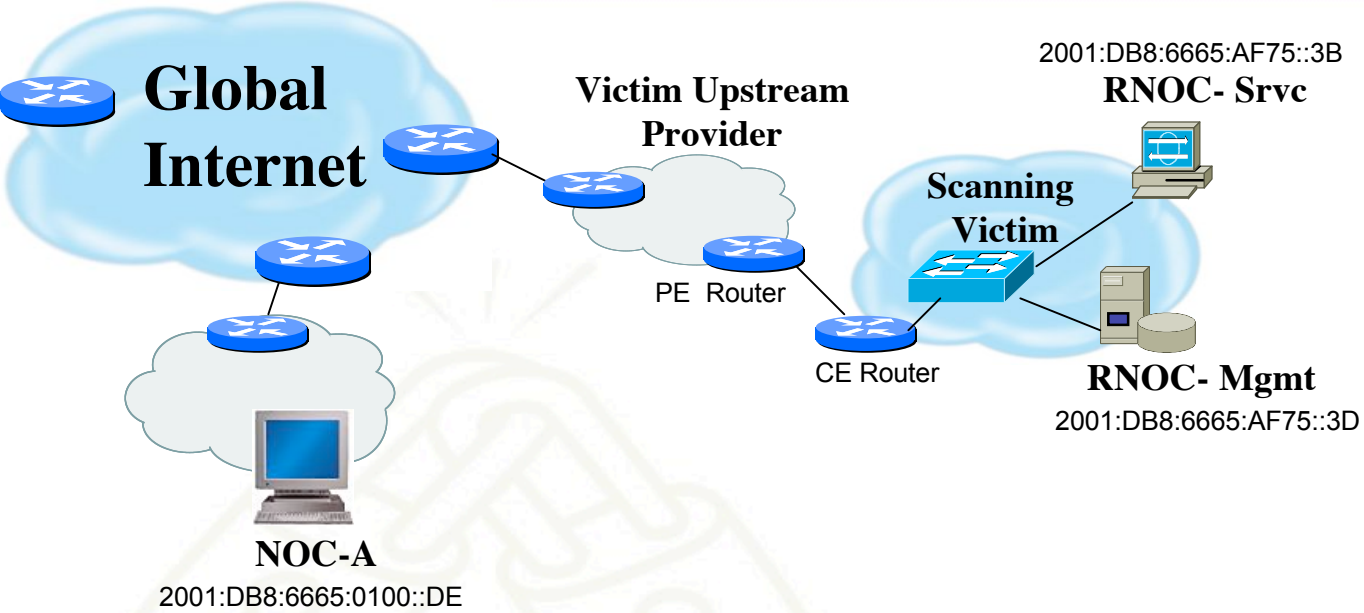


# Protecting Against Scanning Attacks

**Attacker**



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



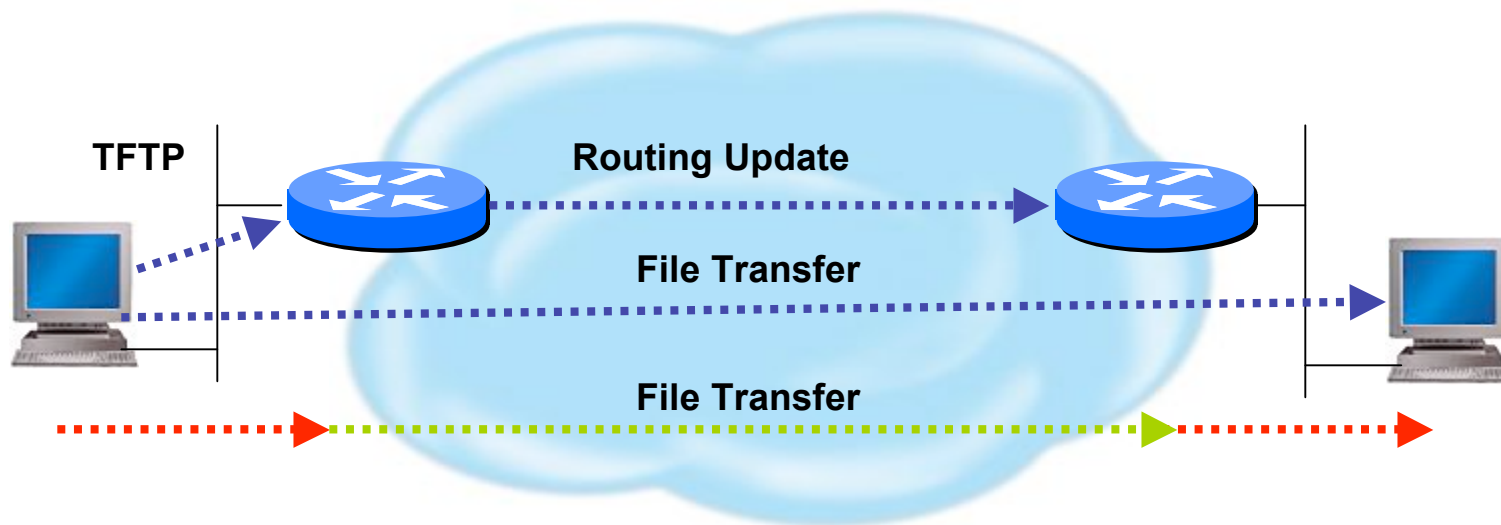
# IPv6 Architectures using IPsec

- 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



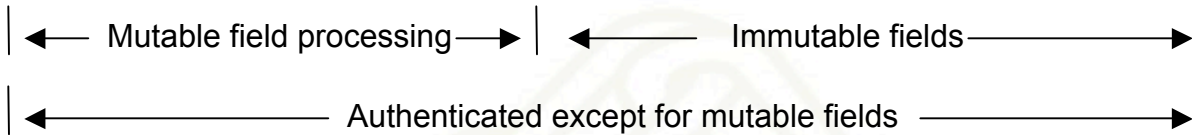
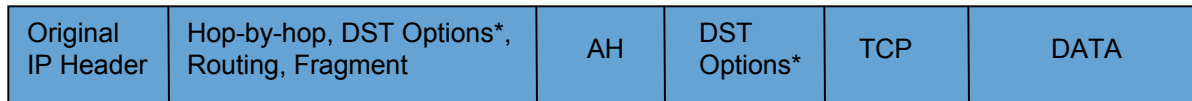
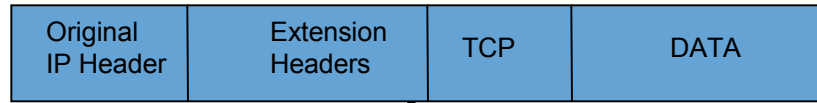
**Transport Mode:** End systems are the initiator and recipient of protected traffic

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



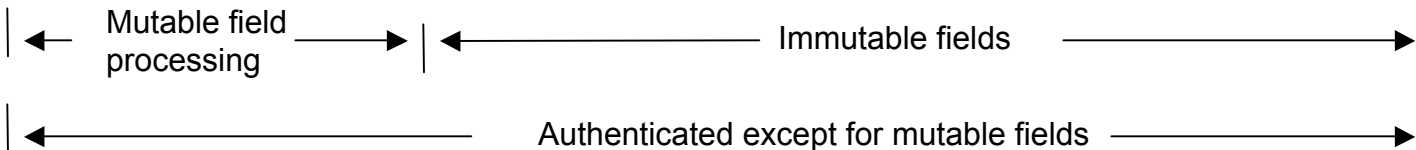
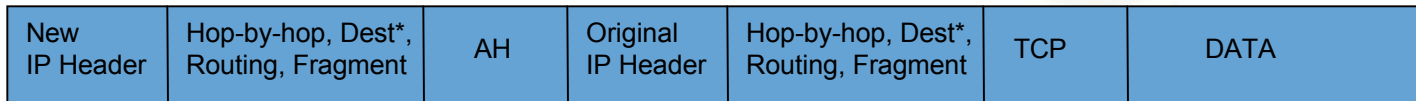
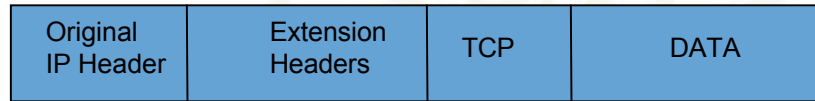
# IPv6 IPsec AH

## IPv6 AH Transport Mode:



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

## IPv6 AH Tunnel Mode:

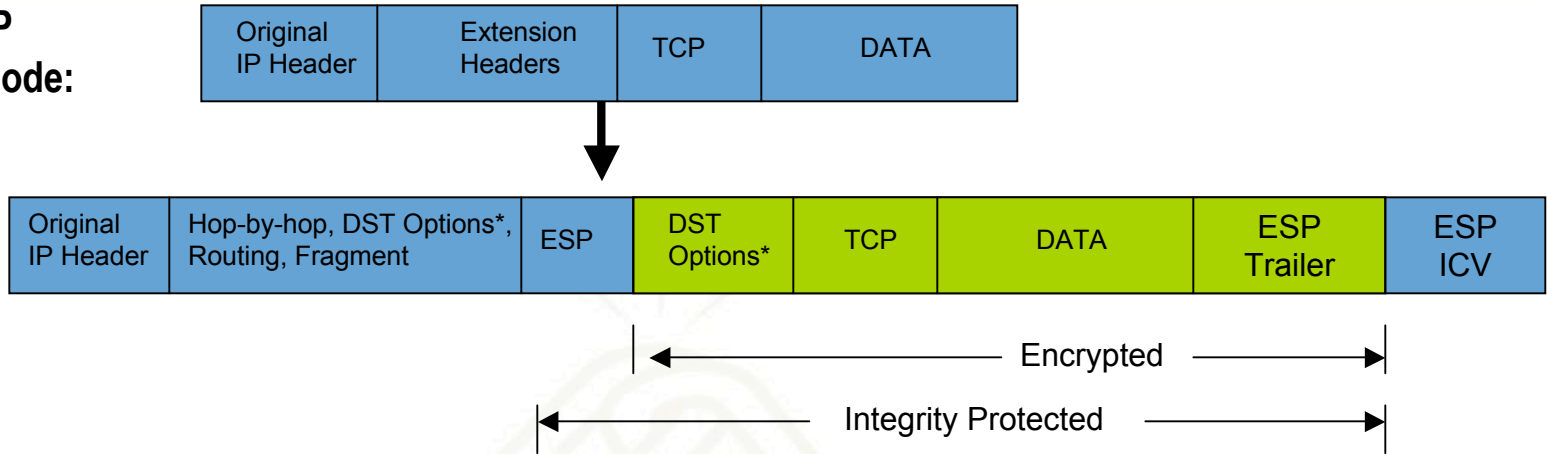


- Mutable Fields:**
- DSCP
  - ECN
  - Flow Label
  - Hop Limit

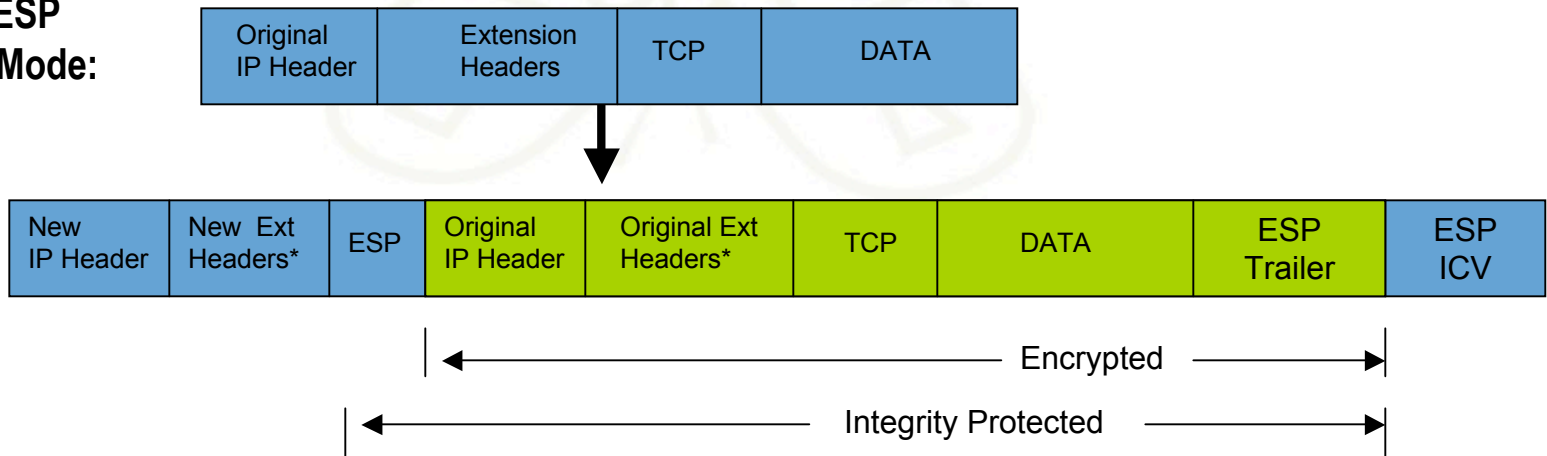


# IPv6 IPsec ESP

## IPv6 ESP Transport Mode:



## IPv6 ESP Tunnel Mode:





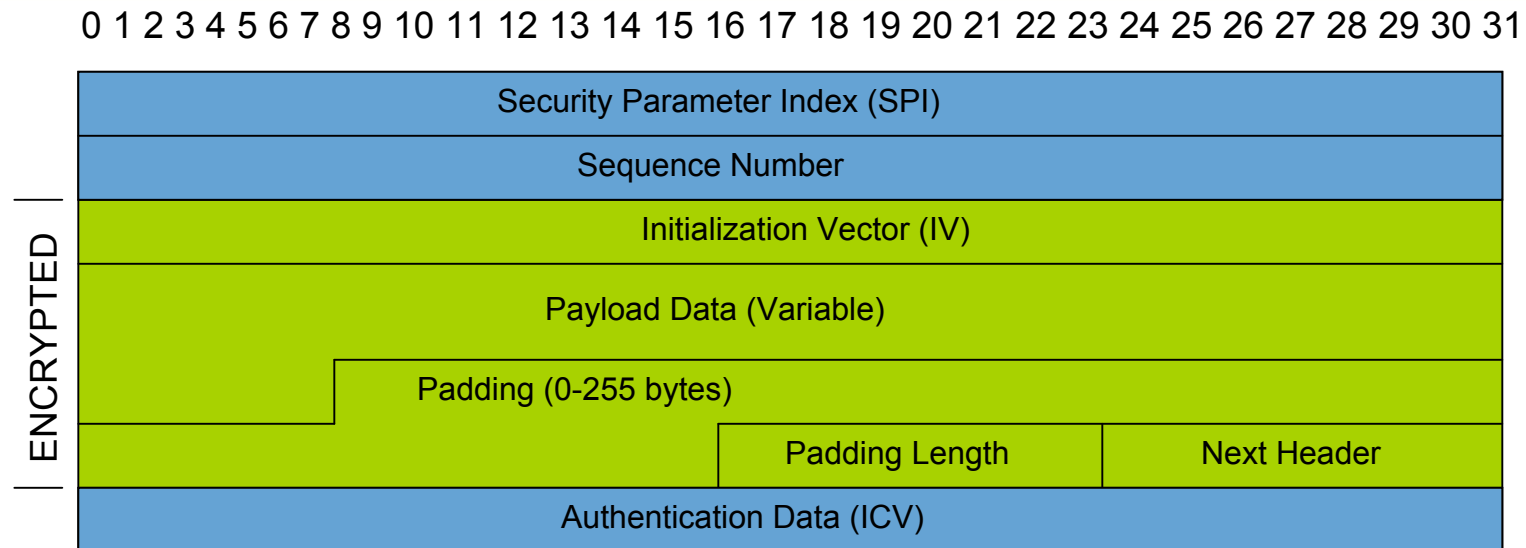


# Enhancements Needed

- 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



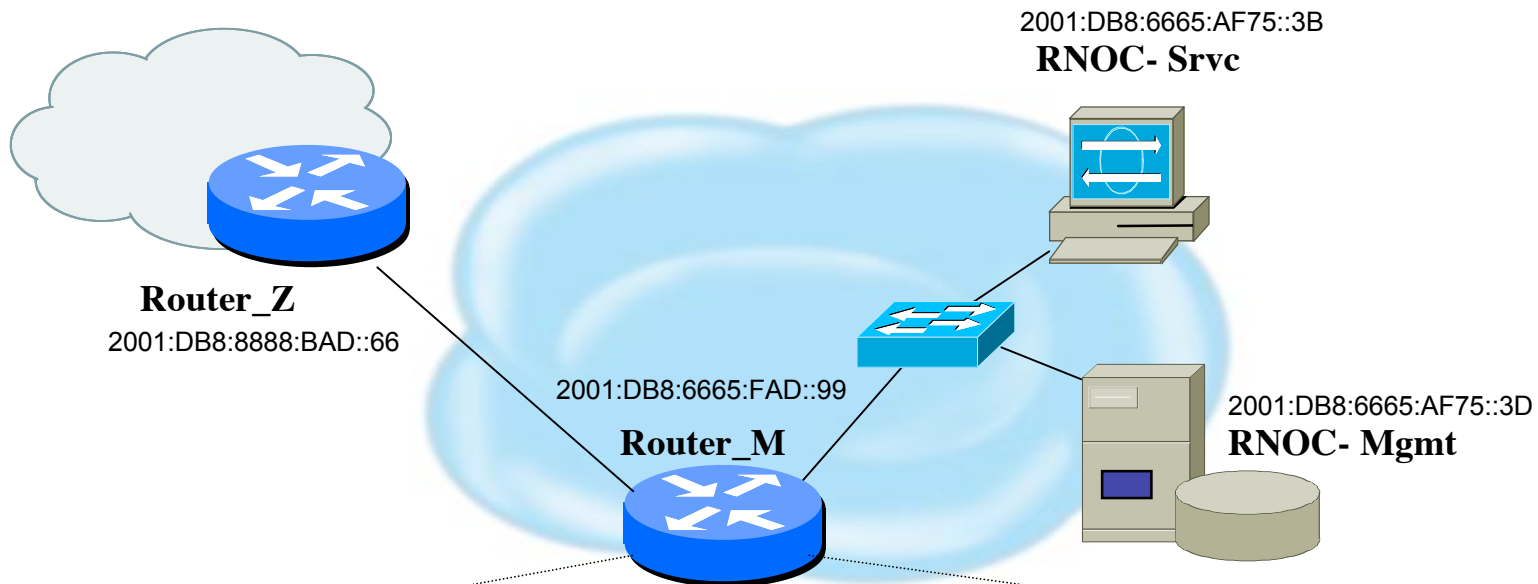
# ESP Header Format



- 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

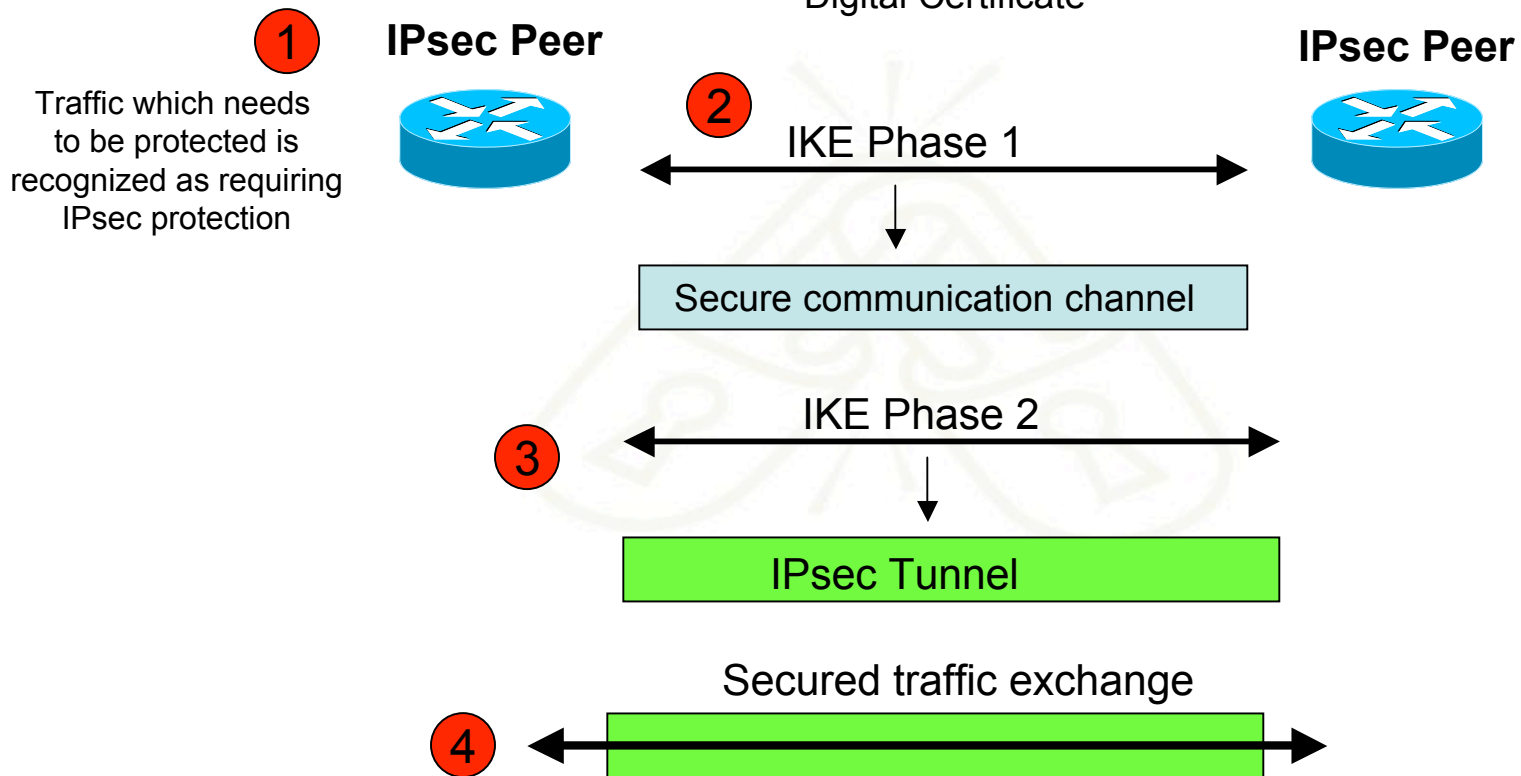
- Phase I
  - Establish a secure communication channel (ISAKMP/IKE SA)
    - Main Mode
      - Negotiates an ISAKMP SA which will be used to create IPsec SAs
      - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DF group to use)
      - Do a Diffie-Hellman exchange
      - 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





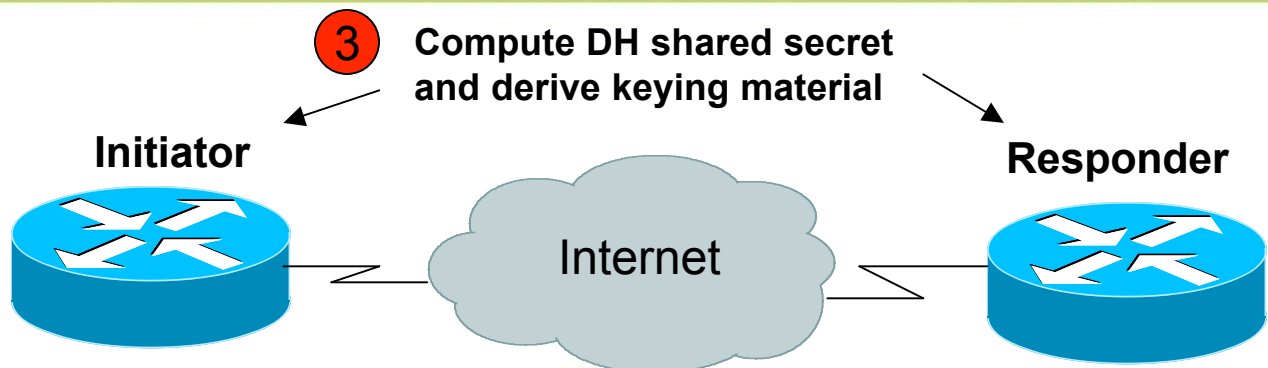
# IPsec IKE Phase 1 Uses DH Exchange

- 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

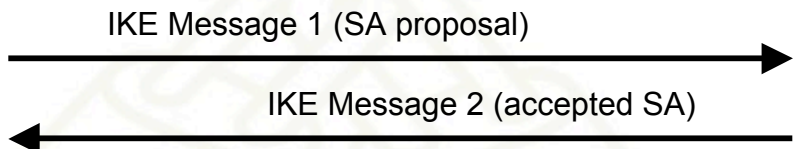




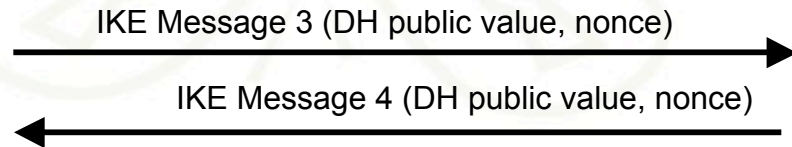
# IKE Phase 1 Main Mode



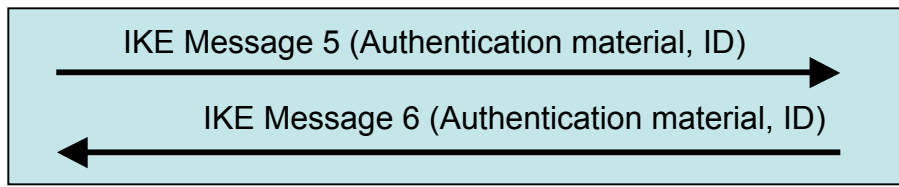
1 Negotiate IKE Policy



2 Authenticated DH Exchange

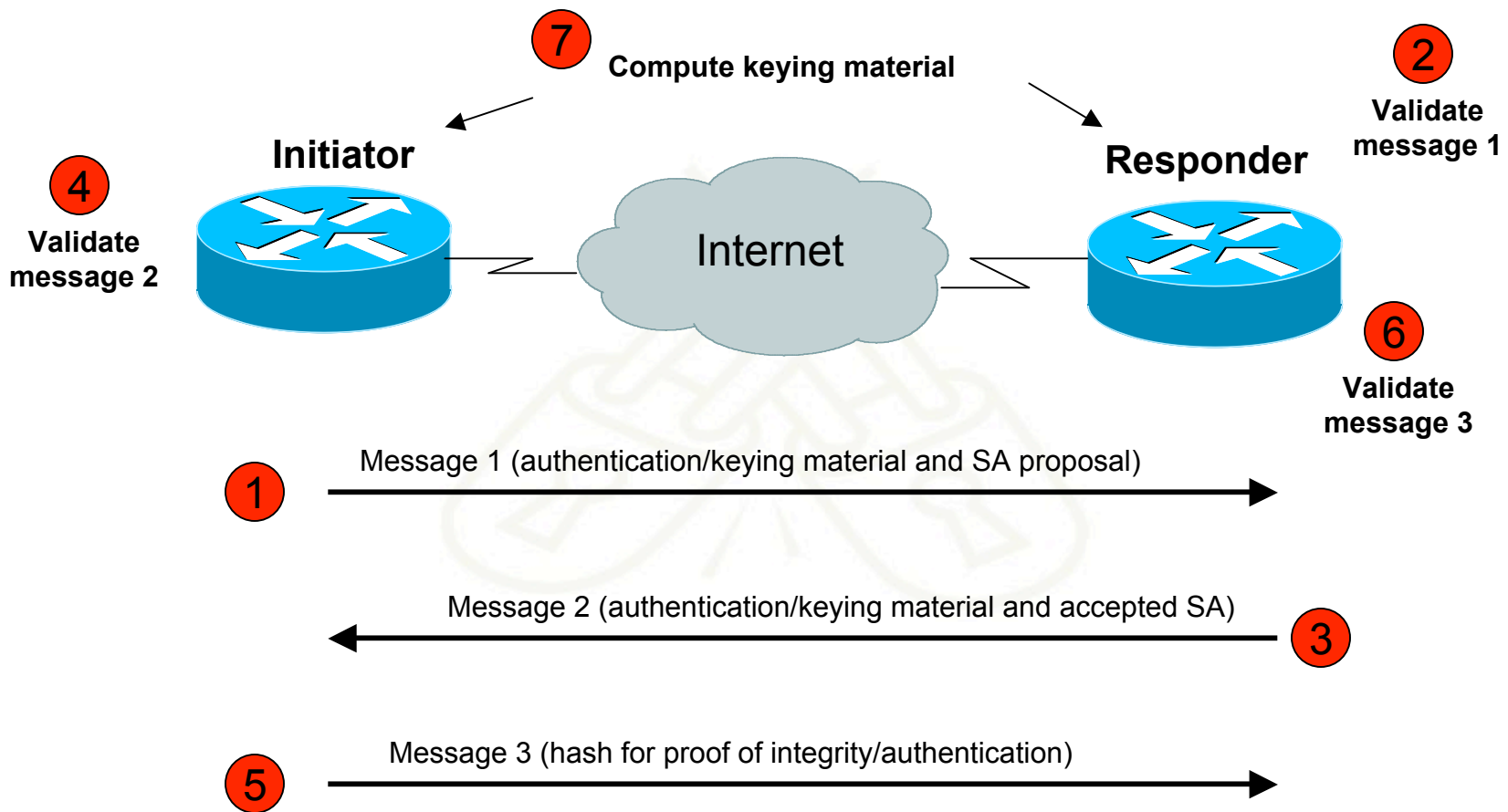


4 Protect IKE Peer Identity





# 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](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/racoon*:
  - ***psk.txt*** – file which contains the shared secrets
  - ***racoon.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/racoon/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 ;
}
```



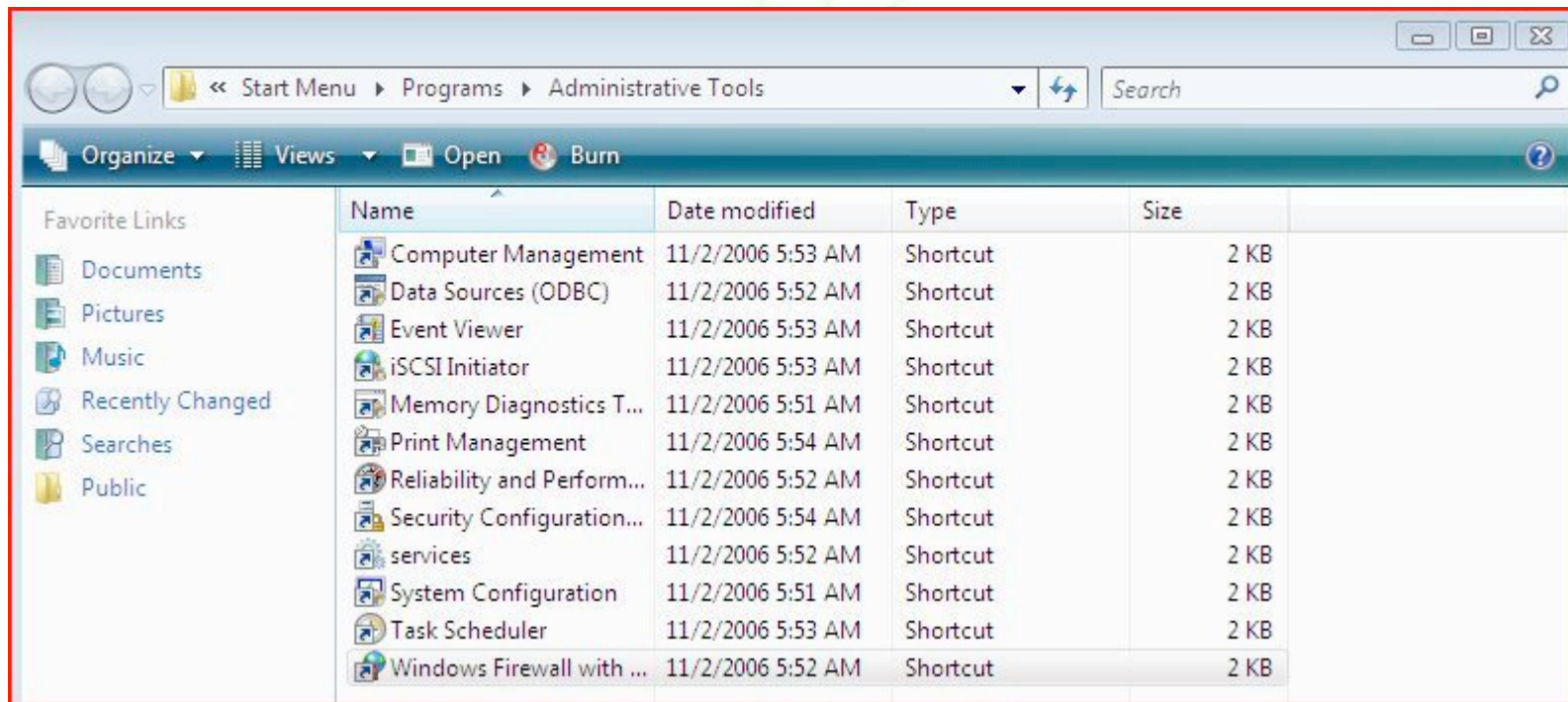
# Testing Racoon

- 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

The screenshot displays the Windows Firewall with Advanced Security console. The main pane shows the status of the firewall on the local computer:

- Windows Firewall is on.
- Inbound connections that do not match a rule are blocked.
- Outbound connections that do not match a rule are allowed.

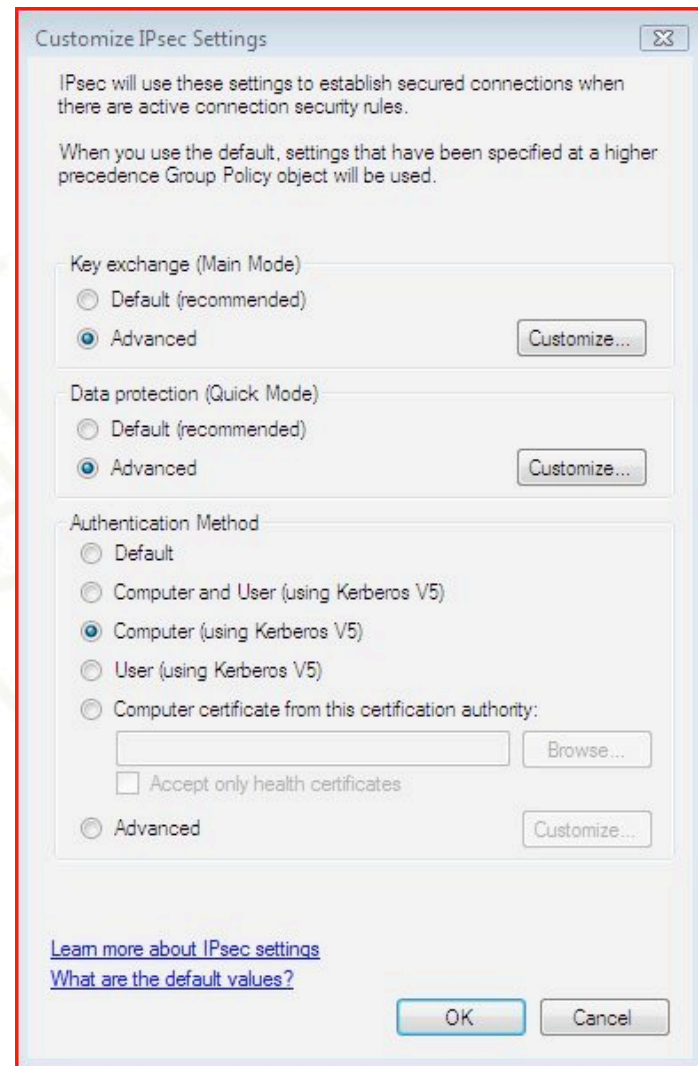
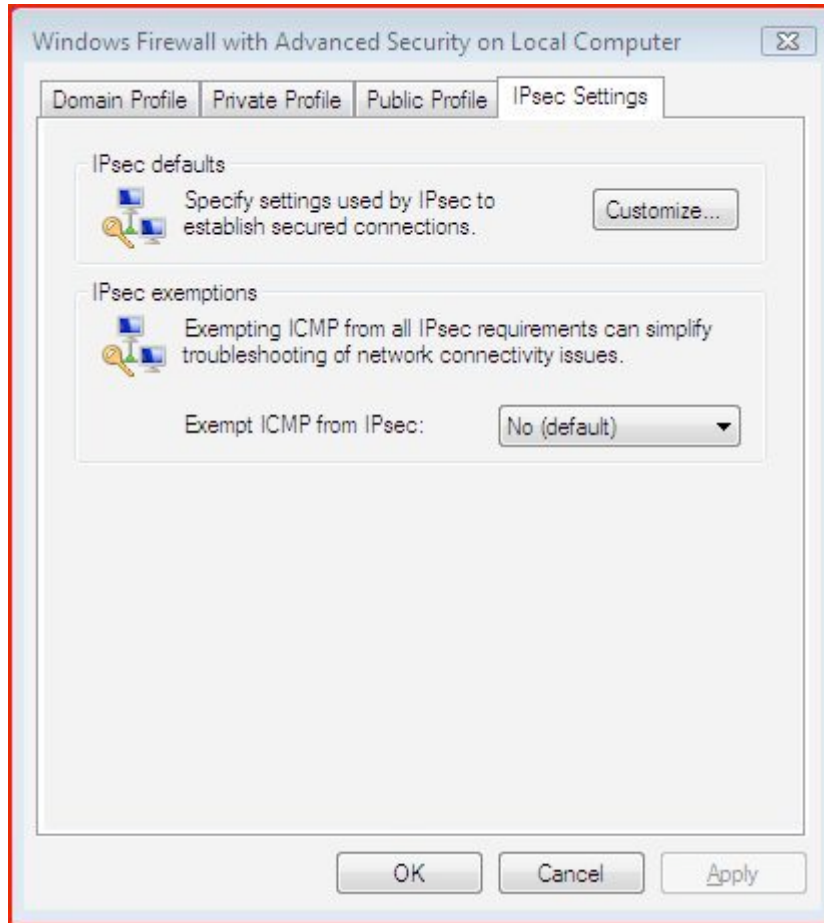
Below this, there are sections for configuration:

- Getting Started:** Authenticate communications between computers. Specify how and when connections between computers are authenticated and protected using Internet Protocol security (IPsec). After specifying how to protect connections using connection security rules, create firewall rules for connections you wish to allow.
  - Connection Security Rules
- View and create firewall rules:** Create rules to allow or block connections to specific programs or ports. You can further restrict connections based on criteria such as whether the connection is authenticated or the users or groups who are initiating the connection. If a connection does not match a specified rule, the default behavior applies.
  - Inbound Rules
  - Outbound Rules
- View current policy and activity:** View information about currently applied policy settings and security associations for active connections.
  - Monitoring

The right-hand pane shows the **Actions** menu with options: Import Policy..., Export Policy..., Restore Defaults, View, Refresh, Properties, and Help.



# Vista: Customizing IPsec Settings





# Vista IPsec Defaults

Windows Firewall with Advanced Security

Hide Back Print Options

Contents Search Favorites

### Default Settings

#### Default settings for Windows Firewall with Advanced Security

These are the default IPsec configuration settings for connection security rules that Windows Firewall with Advanced Security uses before any configuration changes are made.

#### Key Exchange

Settings	Value
Key lifetime (minutes)	480 minutes
Key lifetime (sessions)	0 sessions*
Key exchange algorithm	Diffie-Hellman Group 2
Security methods (integrity)	SHA1
Security methods (encryption)	AES-128 (primary)/3-DES (secondary)

\*A session limit of zero (0) causes rekeys to be determined only by the **Key lifetime (minutes)** setting.

#### Data Integrity

Setting	Value
Protocol	ESP (primary)/AH (secondary)
Data integrity	SHA1
Key lifetimes	60 minutes/100,000 KB

### Default Settings

Setting	Value
Protocol	ESP
Data integrity	SHA1
Data encryption	AES-128 (primary)/3-DES (secondary)
Key lifetimes	60 minutes/100,000 KB

#### Data encryption

Setting	Value
Protocol	ESP
Data integrity	SHA1
Data encryption	AES-128 (primary)/3-DES (secondary)
Key lifetimes	60 minutes/100,000 KB

#### Authentication Method

By default, computer Kerberos (Kerberos version 5 authentication) is used as the authentication method.

#### How default settings work with Group Policy

Policies created using the Windows Firewall with Advanced Security snap-in and distributed with Group Policy, are applied in this order of precedence:

1. Highest precedence Group Policy object (GPO)
2. Dynamic
3. Local
4. Service defaults (if no other defaults are configured)



# Vista: Customizing Data Protection

Customize Data Protection Settings ⌵

Data protection settings are used by connection security rules to protect network traffic.

Require encryption for all connection security rules that use these settings.

**Data integrity**  
Protect data from modification on the network with these integrity algorithms. Those higher in the list are tried first.

Data integrity algorithms:

Protocol	Integrity	Key Lifetime (minutes/KB)
ESP	SHA1	60/100,000
AH	SHA1	60/100,000

**Data integrity and encryption**  
Protect data from modification and preserve confidentiality on the network with these integrity and encryption algorithms. Those higher in the list are tried first.

Data integrity and encryption algorithms:

Protocol	Integrity	Encryption	Key Lifetime (min...)
ESP	SHA1	AES-128	60/100,000
ESP	SHA1	3DES	60/100,000
AH and ...	SHA1, ...	AES-256	60/100,000

[Learn more about integrity and encryption](#)  
[What are the default values?](#)

OK Cancel



# Vista: Customizing Key Exchange

Customize Advanced Key Exchange Settings

**Security methods**  
Use the following security methods for key exchange. Those higher in the list are tried first.

Security methods:

Integrity	Encryption	
SHA1	AES-128	
SHA1	3DES	

↑  
↓

Add... Edit... Remove

**Key lifetimes**  
Determine when a new key is generated. If both options are selected, a new key is generated when the first threshold is reached.

Key lifetime (in minutes):

Key lifetime (in sessions):

[Learn more about key exchange settings](#)  
[What are the default values?](#)

**Key exchange algorithm**

- Elliptic Curve Diffie-Hellman P-384**  
Strongest security, highest resources usage. Compatible only with Windows Vista and later systems.
- Elliptic Curve Diffie-Hellman P-256**  
Stronger security, medium resource usage. Compatible only with Windows Vista and later systems.
- Diffie-Hellman Group 14**  
Stronger than DH Group 2.
- Diffie-Hellman Group 2 (default)**  
Stronger than DH Group 1.
- Diffie-Hellman Group 1**  
This algorithm is provided for backward compatibility only.

OK Cancel





# Vista: Customizing Authentication

Customize Advanced Authentication Methods

**First authentication**  
Specify computer authentication methods to use during IPsec negotiations. Those higher in the list are tried first.

First authentication methods:

Method	Additional Information
Computer (Kerberos V5)	

Buttons: Add..., Edit..., Remove

First authentication is optional

[Learn more about authentication settings](#)  
[What are the default values?](#)

**Second authentication**  
Specify user authentication methods or a health certificate to use during IPsec negotiations. Those higher in the list are tried first.

Second authentication methods:

Method	Additional Information
--------	------------------------

Buttons: Add..., Edit..., Remove

Second authentication is optional

A second authentication cannot be specified if a preshared key is in the first authentication.

**First Authentication Method**

Select the credential to use for first authentication:

- Computer (Kerberos V5)
- Computer (NTLMv2)
- Computer certificate from this certification authority (CA):
- Accept only health certificates
- Enable certificate to account mapping
- Preshared key (not recommended):

Preshared key authentication is less secure than other authentication methods. Preshared keys are stored in plaintext. When preshared key authentication is used, Second Authentication cannot be used.

[Learn more about the first authentication method](#)

Buttons: OK, Cancel



# 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