Global Information Assurance Certification Paper

Windows 2000 Implementation Examples using IPSEC

SANS/GIAC Level 2 Windows Security Practical
Baltimore, MD – 2001

Rinaldo Ribeiro

Table of Contents

1. Introduction

This is a practical assignment for "Securing Windows" SANS' Track, attended at Baltimore, MD on May 2001 and it was written to complete requirements for GIAC Certification in NT Security.

What this is all about?

There are many nice new features at Windows 2000 like WFP, Windows File Protection and Kerberos authentication that its predecessor operating system didn't have. IPSEC implementation can be considered an important new feature that protects resources and information in windows 2000 environments. Its implementation trough group policy distribution or single server installation can be a fundamental step to improve the security.

The main idea here is to demonstrate how the IPSEC implementation used in Windows 2000, which was developed with Cisco, can be deployed and how it can help to improve the security in windows environments.

This paper will also try to apply all the concepts demonstrated to real-life situations and cases where the demand for confidentiality, integrity and strong authentication were premises. It will help you to protect your information using IPSEC to create VPNs and packet filters, controlling access to your resources. It will also guide you using examples and real implementations.

This document assumes that you are familiar with the Internet Protocol and general security terms and concepts. IT professionals, network admins or any person interested in IPSEC implementation using Windows 2000 can use the information included.

2. Understanding IPSEC

Before we jump into the real-life scenario and implementation examples it's important to understand what is really IPSEC, how it works, which protocols are used and how they are used to create VPNs and implement security through packet filters.

Originally described in RFCs 1825-1829, written in 1995, IPSEC was designed to provide security to versions 4 and 6 of IP protocol, including access control, integrity, authentication and confidentiality. These objectives are met through the use of two protocols, the Authentication Header (AH) and the Encapsulating Security Payload (ESP), and through the use of cryptographic key management procedures and protocols.

The AH protocol offers authentication of packet source, data integrity and protection against replay attacks while ESP provides all these services plus data encryption. Its important to notice that ESP protocol doesn't substitute AH and both can be used simultaneously. The main reason is that AH authenticates the IP layer and higher while ESP only authenticates the transport layer and higher. It means that ESP doesn't detect changes in the IP layer of the packet.

Any IP protocol can be used over IPSEC, trough encrypted tunnels (VPNs) or just to implement encryption between computers. In other words this is called tunnel mode and transport mode. It's quite common to see these terms around. Tunnel mode means that the entire packet will be encapsulated in a new header. This is a common situation of encryption between two sites or two different networks. When two gateways are used to create a VPN using IPSEC between two different networks, tunnel mode is used. Transport mode keeps the original header and is usually used to encrypt and/or authenticate packets (depending on which protocol is used: AH, ESP or both) between computers directly.

So AH and ESP could be used in transport or tunnel mode. Just to give you an idea of how the packets are changed, take a look at these examples:

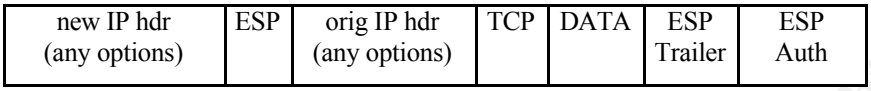
Original packet:

orig IP hdr (any options)	TCP	DATA

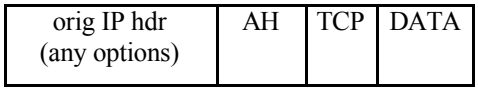
ESP in transport mode:

orig IP hdr (any options)	ESP hdr	TCP	DATA	ESP Trailer	ESP Auth

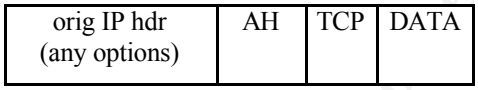
ESP in tunnel mode:



AH in transport mode:



AH in tunnel mode:



3. IKE/ISAKMP

IKE is defined as a hybrid protocol that is used to negotiate and provide authenticated keying material for security associations (SA) in a protected manner, according to RFC 2409. It is a subset implementation of ISAKMP/Oakley protocols, and that's why sometimes these terms are misunderstood. Before two peers start exchanging encrypted data using IPSEC, IKE is used to agree on how to secure their communication.

The concept of a "Security Association" (SA) is fundamental to IPSEC. A SA is a result of an IKE negotiation and it contains security parameters such as shared session key, identity authentication method, data authentication and encryption algorithm. Each SA has one SPI or Security Parameter Index as a unique identification number. Each IPSEC-enabled host will keep the SAs in a database called SADB or Security Association Database. Both AH and ESP make use of SAs and a major function of IKE is the establishment and maintenance of Security Associations.

IKE uses two phases to negotiate the security parameters and each phase negotiates a different type of SA. Phase I negotiates what is called "IKE-SA". This security association will be used to establish the second type of SAs, the "IPSEC-SA". An IKE-SA must exist to an IPSEC-SA be generated.

In Phase I:

- ✓ A policy is negotiated (encryption algorithm, hash algorithm, authentication method and Diffie-Helman group)
- ✓ A Diffie-Helman key exchange is performed
- ✓ Peers are authenticated
- ✓ IKE-SA is created.

In Phase II:

- ✓ A policy is negotiated (IPSEC protocol - AH - ESP, encryption algorithm and hash algorithm)
- ✓ Key generation (Master key from phase I is used or a new key is generated for Perfect Forward Secrecy)
- ✓ IKE-SA is created and stored in the SADB with an unique SPI

The security offered by IPSEC ultimately depends on the quality of its implementation, which means the users must trust that a vendor implemented correctly the standards described on the RFCs. IKE implementation in Windows 2000 was "jointly developed by Microsoft and Cisco Systems" according to "advanced" tab located inside an IPSEC policy.

Since the IPSec framework is standardized, an implementation is never vendor dependent. It can be applied to routers, firewalls, client desktops and Windows 2000 boxes.

4. Windows 2000 IPSEC Implementation

IPSEC on Windows 2000 is totally integrated with the operating system itself. By default a Windows 2000 machine can implement a local IPSEC policy without depending on any external element or third party software. But if it is part of a domain, an IPSEC policy can be implemented and managed through Group Policies. Keep in mind that a local policy will be always overwritten by a policy managed using Group Policy.

Let's define the key elements of this puzzle:

IPSEC Policy Objects

Define all the options necessary to implement IPSEC on a computer such as authentication methods and ciphers. A policy must be "assigned" to a computer to start working and it does **not** require a reboot. :)

Group Policy

It is a way of distributing IPSEC policies. The flexibility of this feature extends the scope of IPSEC. Group policy can be used to many other different reasons not covered by this document.

IPSEC Policy Agent Service

This service implements the IKE protocol and it is what retrieves, interprets and enforces IPSEC Policies. It is listed in the Services applet.

IKE Protocol

It is the negotiator. Security parameters necessary to establish the communication are negotiated between two peers using IKE protocol. (Don't forget that IKE is an implementation of ISAKMP/Oakley protocol. It is a simplified version.)

IPSEC Driver

The IPSEC Driver, ipsec.sys, uses the filters and filter actions defined in the policy assigned. The policy agent is the manager and the IPSEC driver is responsible for handling packets.

Security Association API

Provides the interface between the IPSec driver, IKE and the Policy Agent.

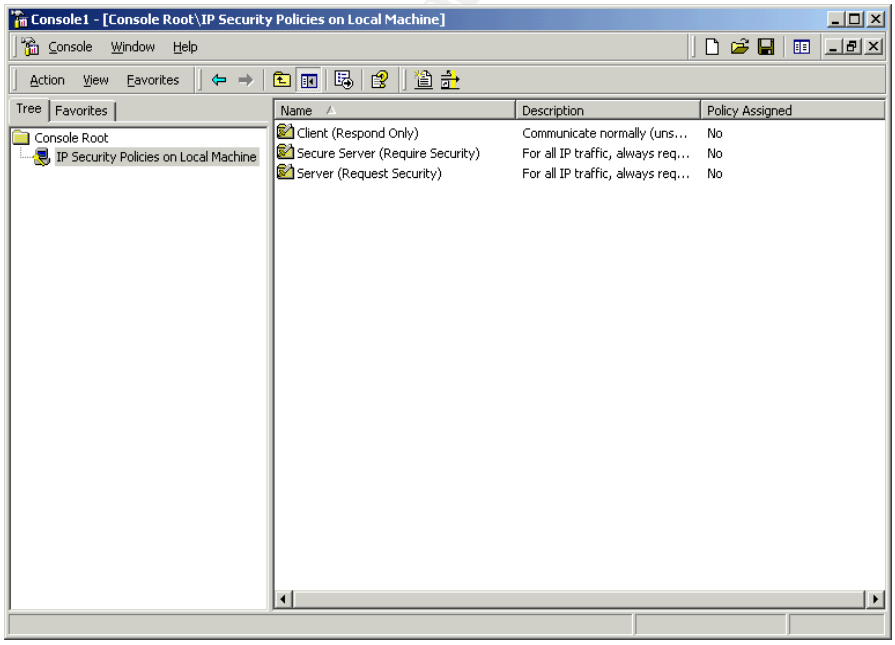
© SANS Institute 2000 - 2005 Author retains full rights.

4. Creating a IPSEC Policy

It is not a difficult task to create an IPSEC policy. Just few mouse clicks and "boom" it's gone. But it is mandatory that you understand what exactly you are doing before you start assigning policies to your network. To create a policy you must know how to create IPSEC rules, IP filter lists and filter actions. It's all about managing rules.

Filter lists are what will trigger the IPSEC driver to deal with certain packets, applying the actions related to the filter. An IPSEC policy contains IPSEC rules that contain IP filter lists and filter actions. Each policy can have multiples rules but each rule can have only one IP Filter list and one filter action. Easy, huh? So, to create and assign an IPSEC policy you have to create and manage IPSEC rules.

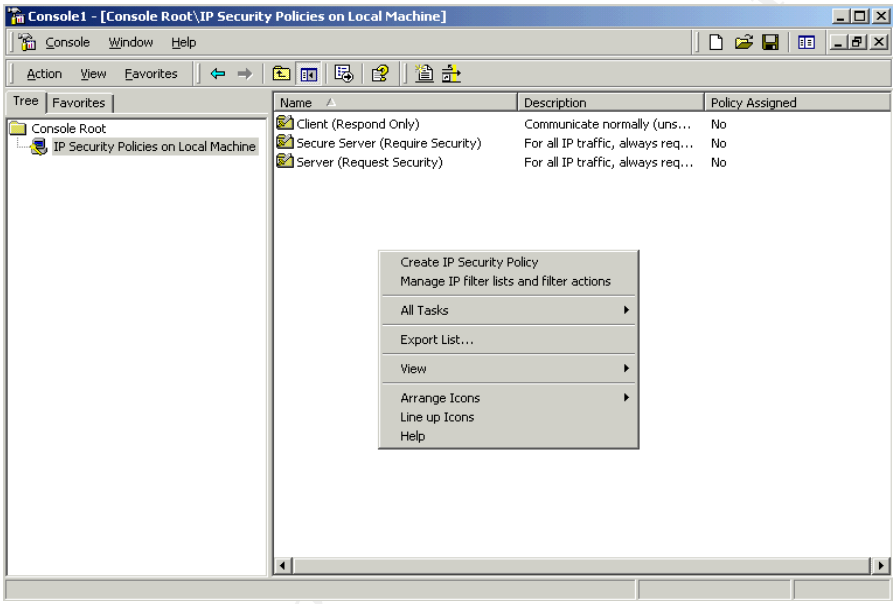
MMC can be used to create IPSEC policies. Try now. Go to Start/Run and type "MMC". Then go to "Console", "Add/Remove Snap in" and than "Add". Now you can choose which snap-ins you want to include to your management console. For the purpose of this context, only include "IP Security Policy Management" selecting it on the list and clicking on "Add". Because we are managing a local computer, leave the default setting "Local Computer" and click "Finish". Then just click "Close" and "OK". Now we are able to start managing IPSEC policies. That is what you should see after following the steps:



Notice that you have three pre-defined policies by default with a description and a "Policy Assigned" column. A policy is assigned by right clicking and selecting "Assign" and the most important, without any reboot. It's recommended to create your own policies instead of using one of the defaults. Creating a new one will be an opportunity to adjust your reality to your policy needs.

IPSEC filter lists and filter actions can be used by many IPSEC rules. So it's up to you to choose a start point. You can create filter lists and actions before the rules or you can create a basic rule and add the filters lists and actions later. For example there is no filter action "block" by default and you will see later on this document how this action can be useful in a packet filter implementation. You could create this action before the rule itself. Again, it is up to you to choose.

When you right click on the right side of MMC, you see "Create IP Security Policy" and "Manage IP Filter lists and filter actions", as the screen bellow:



The whole process of creating and implementing an IPSEC rule will be demonstrated on the next item "Implementation Examples". For this moment it's important to notice how IKE Phase's settings are configured.

If you double click on a default policy for example you will see a properties screen with two tabs: "Rules" and "General". Each tab is related to a phase. The "General" tab contains all the settings related to Phase I, which means that it's possible to configure key exchange settings, PFS – perfect forward secrecy and security methods such as encryption algorithms and Diffie-Hellman group. The "Rule" tab contains all settings related to Phase II and it's possible to configure IP filter lists, filter actions, authentication and tunnel settings.

IPSEC processing summary:

- A packet matches an IP Filter
- The IPSEC driver requests that IKE negotiate a security method and a security key.
- IKE negotiates a security method and sends it with a security key to IPSEC driver.
- The IPSEC driver stores these settings, part of the security association - SA, in its database.

- Both peers know and store the SA to decide which traffic should be protected.

5. Implementation examples

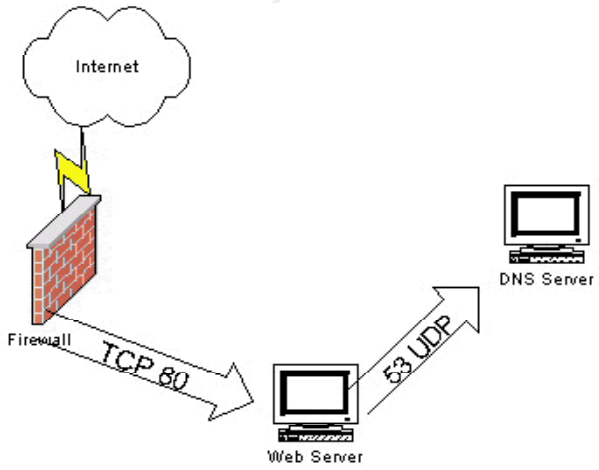
Now is the time that everyone was waiting for. It is time to see implementation examples and real scenarios.

5.1 - Protecting a web server using IPSEC filters

It is already known that IPSEC policies can help to protect a Windows environment. What about "IPSEC packet filtering"? With Windows 2000 machines it is possible to create packet filters using IPSEC implementation. Instead of configuring encryption settings, you can use policies to implement packet filters. IPSEC rules are responsible for triggering the driver to act. Permit and deny is a question of action.

Let's suppose you have an IIS 5.0 running on a windows 2000 server. The only connection allowed in your firewall is to port 80 TCP to this server. What you could do without touching your firewall configuration is to use IPSEC to create packet filters locally in your server.
It is in fact a way to double enforce access control to a resource, in this case your web server, but don't forget that internal people can represent a threat also.

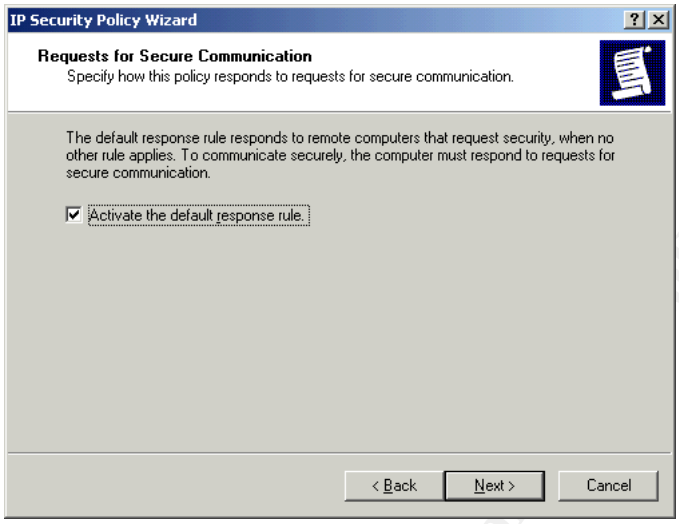
For this case the web server is running in a dedicated machine and the only traffic allowed from the server itself is to access DNS servers. Inbound traffic is permitted only to TCP 80. The policy will allow the external world to access only port 80 TCP and permit connections to DNS servers.



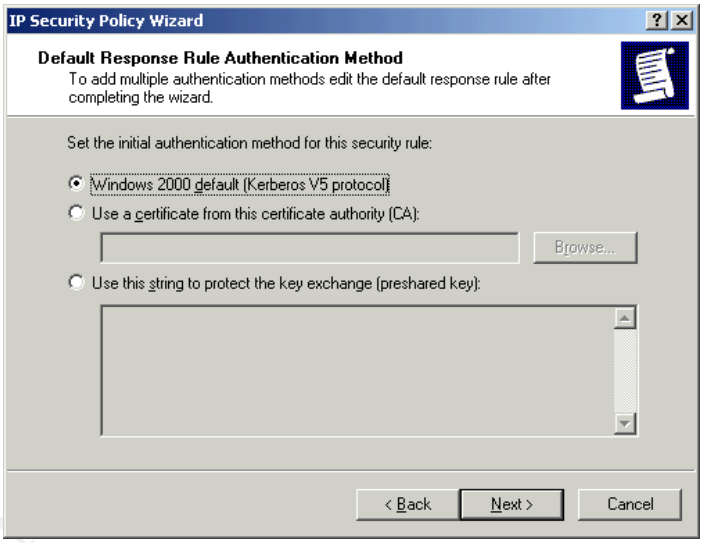
It is important to create a new action "block" that by default doesn't exist. It makes sense. If you think that a filter will trigger an action to protect the traffic and if you were blocking packets, why protect them? Anyway, let's create the policy and then populate it with filter lists and actions.

To create the policy, right click on the right side of MMC screen and select "Create IP Security Policy". The wizard will be loaded to help you. Click next on the first screen and then choose a name and a description for your policy. The next screen is about the "default response rule". This rule, if activated, will be the response rule to computers that request security when no other

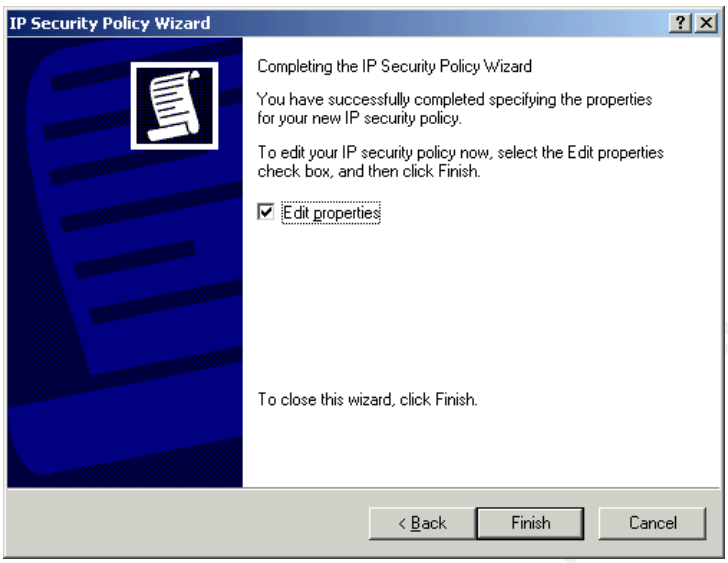
existent rule applies. For now, leave this option selected and click next.



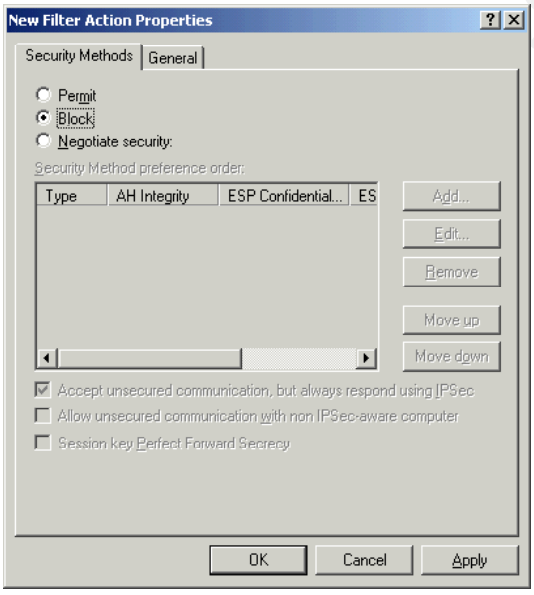
Because we are creating a packet filter there is no need for authentication. So this option doesn't matter to this case. Just click next.

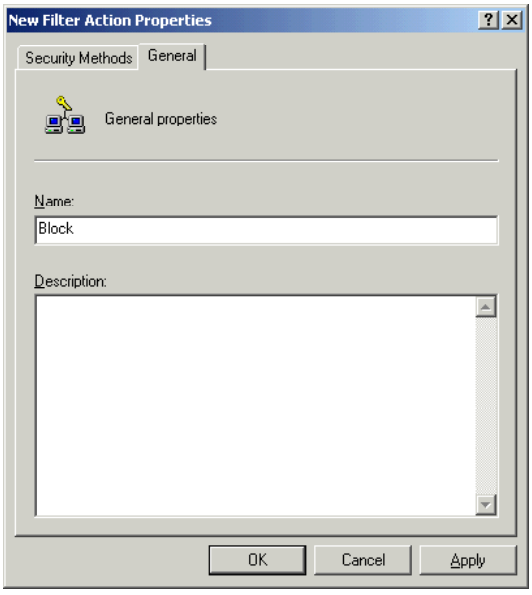


The screen bellow indicates that you finished specifying properties for you policy. Leaving the box "Edit properties" marked will send you to the same screen if you double click on a policy. Uncheck this option. New filter lists and action must be created before finishing the policy.

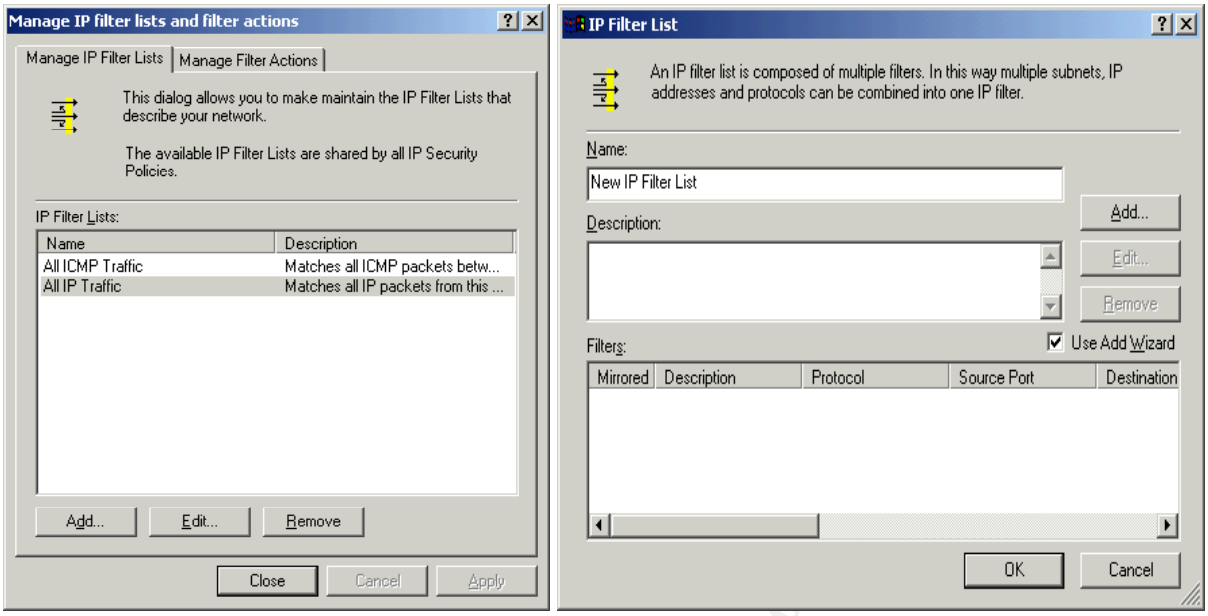


Now you have to create filter lists and add a new action. To do this go to "Manage IP filter lists and filter actions" following the same steps described when you create the new policy. The right tab is "Manage Filter Actions". Click on "Add" and select "Block". Before clicking on "Ok", go to "General" tab and change the default name to "Block". You have just created a new filter action.





To create new filter lists, click on "Manage IP Filter Lists". It's the left tab on the same screen that "Manage Filter Actions" is. Click on "Add", select a name and a description to the filter list then click on "Add" again.



Now it's time to create a proper filter related to our case. After clicking on "Add", each screen will ask you for settings about your rule. The settings are:

- ⇨ IP Traffic Source: "Any IP Address"
- ⇨ IP Traffic Destination: "My IP Address"
- ⇨ IP Protocol Type: "TCP"
- ⇨ IP Protocol Port: "From any port" / "To this port: 80"

This filter will allow only access to port 80 TCP and it will be related to an action when we finish this policy. Each filter created can be used later in a policy.

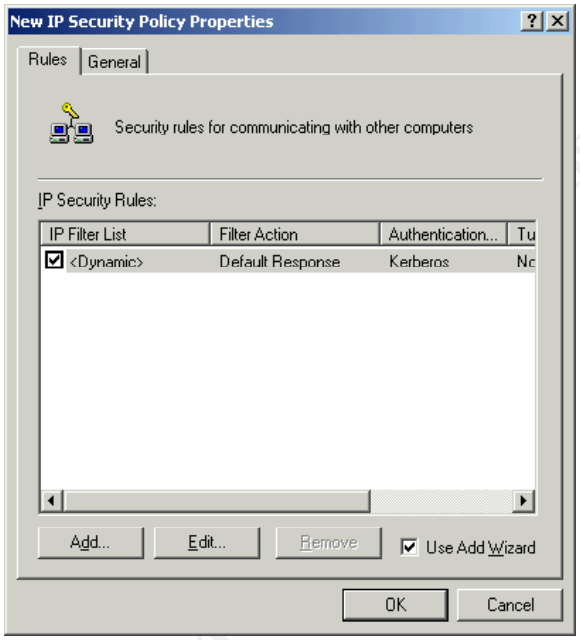
There is an option to set a rule as "mirrored". This setting will create an automatic rule with exact opposite source and destination addresses. The order of the filters doesn't matter. If there is a rule and a packet match this rule, the action will be triggered. But don't forget that a rule can have two different directions and then a packet can match two different rules in two different directions. The IPSEC driver matches packets against filters from the most specific to the least specific.

Let's create a second rule following the same procedures used before to allow access from the web server to DNS servers. Don't forget to name the rule with something related to what it will do, for example "DNS access". These names will help you manage the rules when you need them in a policy. The settings are:

- ⇨ IP Traffic Source: "My IP Address"
- ⇨ IP Traffic Destination: "A specific IP Address"/ DNS_IP_ADDRESS
- ⇨ IP Protocol Type: "UDP"
- ⇨ IP Protocol Port: "From any port" / "To this port: 53"

Now we have all the rules available to finish the policy creation. It is important to notice that after including these rules to allow traffic, another rule will be necessary to block all the other possible traffic. For this reason a default IP filter list will be used ("All IP Traffic") together with the action "block" that was just created.

To finish the policy creation, double click on the policy you created before. The following screen will appear:



Click on "Add" to add filter lists to the policy. There are some settings that will not be used such as authentication parameters and tunnel endpoint. Remember that we are creating a "packet filter IPSEC" and there is no encryption. The idea is control access to the web server. So leave these setting as default. It will not interfere in the policy.

Tunnel Endpoint

- ⇨ Leave this setting as default, "This rule does not specify a tunnel".

Network Type

- ⇨ You can also leave this as default, "All network connections" but you could use "Local Area Network" since we are not using remote connections.

Authentication Method

- ⇨ Leave this setting as default, "Windows 2000 default". Again, this setting does not matter to this case.

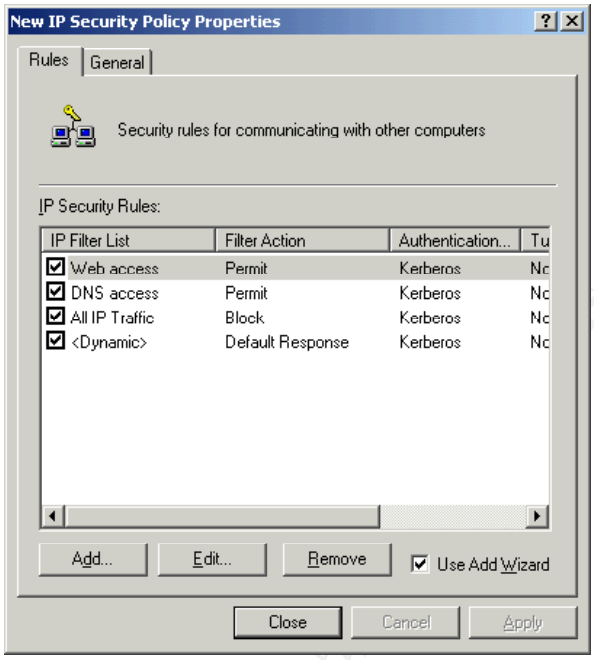
IP Filter list

- ⇨ Select a filter list that was created before like "DNS Access". The order is not important so choose anyone.

Filter Action

- ⇨ Select an action. In this case, "Permit" must be choosen.

Repeat this steps to “Web Access” filter list using the action “Permit” and then use the default filter list “All IP Traffic” with action “Block”. Now we have a policy that will do exactly what we need:



Remember that “General” tab is about Phase I and because this policy will work as a packet filter only, there is no need to configure these settings.

Click on “Close” then right click the policy and select “Assign”. Now there is only two ways of accessing your web server: inbound 80 TCP and outbound UDP 53.

An IPSEC policy will never substitute all the protection that a Windows server requires. It is fundamental to configure the OS in a proper manner and harden the box as much as possible.

5.2 - Secure communication between two Windows boxes

If you have more than one windows 2000 machine in your network and there is a need of confidentiality, authentication and integrity to any communication between them, there is no excuse for not using IPSEC.

On this case two windows 2000 machines, one Professional and another Server, that don't belong to the same or trusted domain, need to securely communicate for a certain reason. The goal is establish a secure connection using IPSEC.

The main concern is authentication. The three possible types of authentication in Windows IPSEC are Kerberos, Certificate and pre-shared secret. Kerberos doesn't apply to this case because it only can be used when the peers are part of the same domain or two trusted domains. Certificate is a good option but requires a trusted CA by both peers and digital certificates for each machine. And finally "pre-shared secret" that is the easiest and weakest option.

Kerberos is always a better option. It is enabled by default and integrated in the OS. Just because we can't use Kerberos is this case, the best option becomes Certificate Authentication.

Before configuring authentication, the policies must be created in both peers. The same steps must be followed to create filter lists but this time it is necessary to configure properly Phase I settings in the "General" tab ("Advanced" button in the policy properties.)

First create a new IP filter list for this policy. Right click on the right side of MMC screen and select "Manage IP Filter lists and filter actions". Click on "Add", select a name and a description to the filter list then click on "Add" again. The settings are:

- ⇨ IP Traffic Source: "My IP Address"
- ⇨ IP Traffic Destination: "A specific IP address"/ OTHER_PEER_IP_ADDRESS
- ⇨ IP Protocol Type: "Any"

This filter list will trigger any IP traffic between the peers.

Create a new policy. Right click on the right side of MMC screen and select "Create IP Security Policy". The wizard will be loaded to help you. Click next on the first screen and then choose a name and a description for your policy. Leave the "default response rule" activated and click "next". Leave the authentication option as default and click "Finish".

On the policy properties screen, click on "Add" at "Rules" tab. The settings are:

Tunnel Endpoint

- ⇨ Leave "This rule does not specify a tunnel" marked. Transport mode will be used.

Network Type

- ⇨ You can also leave this as default, "All network connections" but you could use "Local Area Network" since we are not using remote connections.

Authentication Method

- ⇨ Leave "Windows 2000 default" marked for now. The certificates must be generated before we select certificate authentication to be used.

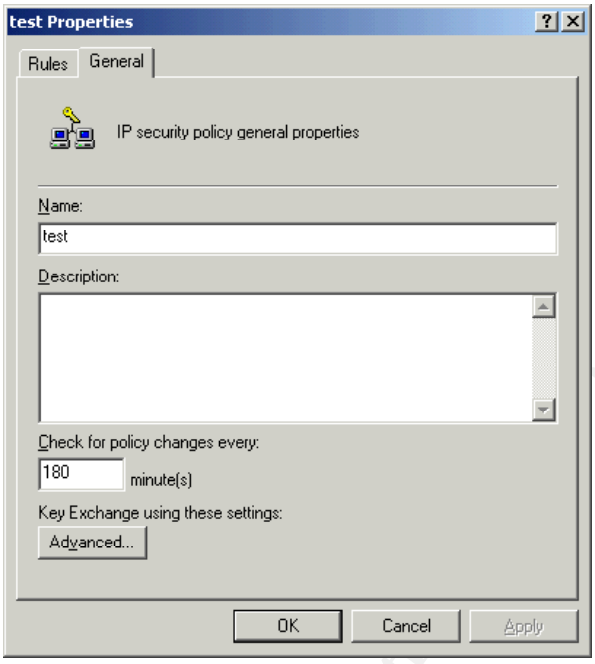
IP Filter list

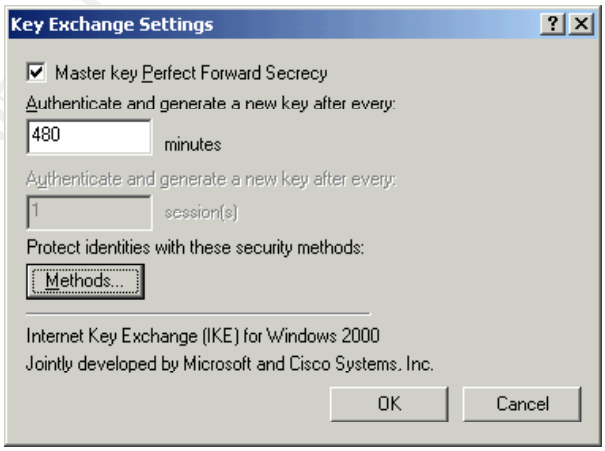
- ⇨ Select the filter list that was created for this connection.

Filter Action

- ⇨ Select the default action "Require Security".

Click on "Advanced" button at "General' tab and select "Master key Perfect Forward Secrecy" for a better security level. The PFS setting will force a new key to be generated for IKE's Phase II, instead of using the key from Phase I.





All these steps must be done in both peers. Observe the IP addresses when creating filter lists to make sure that the right one is used.

Before assigning the policies, digital certificates have to be installed in both peers together with the CA certificate. An external trusted CA must be generate both certificates. You can use a Windows 2000 PKI to generate them or you can use a test CA from Microsoft.

The procedures described bellow about generating digital certificates for IPSEC were taken from the document "How to Install a Certificate for Use with IP Security" from Microsoft's knowledge base. (http://support.microsoft.com/support/kb/articles/Q253/4/98.ASP)

Let's suppose you don't have a CA running in a Windows 2000 server on your local network. No problem. Point your browser to http://sectestca2.rte.microsoft.com/certsrv/ and request your certificates.

Connect to http://sectestca2.rte.microsoft.com/certsrv/ and follow these steps:

1. In the initial Welcome screen of the Certificate server, click **Request a certificate**, and then click **Next**.

2. In the "Choose Request Type" screen, click **Advanced request**, and then click **Next**.

3. In the "Advanced Certificate Requests" screen, click **Submit a certificate request to this CA using a form**, and then click **Next**.

4. In the "Advanced Certificate Request" screen, type your name and your e-mail name in the appropriate boxes.

5. Under **Intended Purpose**, select **Client Authentication Certificate** or **IPSec Certificate**. If you choose **IPSec Certificate**, then this certificate will only be used for IPSec.

6. Under **Key Options**, click **Microsoft Base Cryptographic Provider v1.0**, **Signature** for **Key Usage** and **1024** for **Key Size**.

7. Leave the **Create new key set** option enabled (you can clear the **Container Name** check box unless you want to specify a specific name), and then click **Use local machine store**.

8. Leave all the other options set to the default value unless you need to make a specific change.

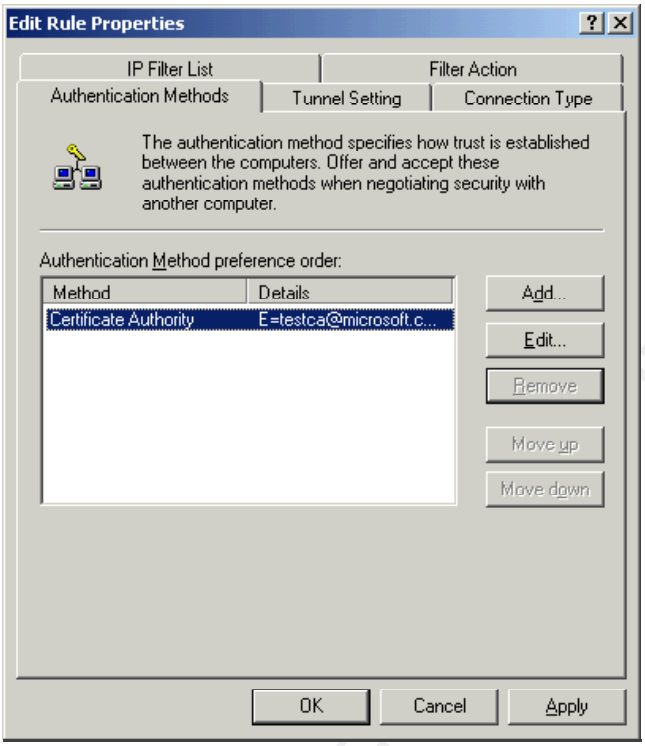
9. Click **Submit**.

10. Click **Install this Certificate**. The "Certificate Installed" screen should appear with the message "Your new certificate has been successfully installed."

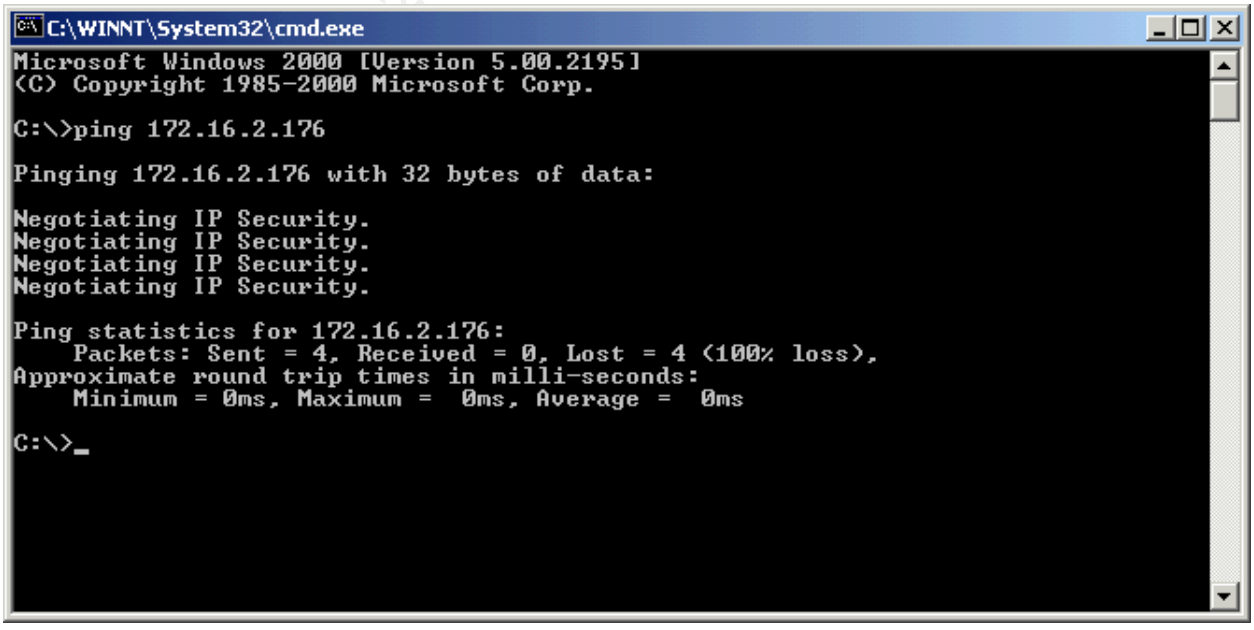
After the certificate is installed, verify the location of the certificate by using the Certificate (Local Computer) snap-in in Microsoft Management Console (MMC). Your certificate should appear under **Personal**.

If the certificate you have installed does not appear here, the certificate was installed as a "User certificate request," or you did not click **Use local machine store** within the advanced request.

Now you can go back to the policy to change the authentication method. Double click you policy, select the proper rule and click “Edit”. Select the tab “Authentication Methods” and click on “Add”. Select “Use a certificate from this certificate authority (CA)” and click on “Browse”. Choose the certificate issued to and by SECTESTCA1. Repeat these steps on both peers.

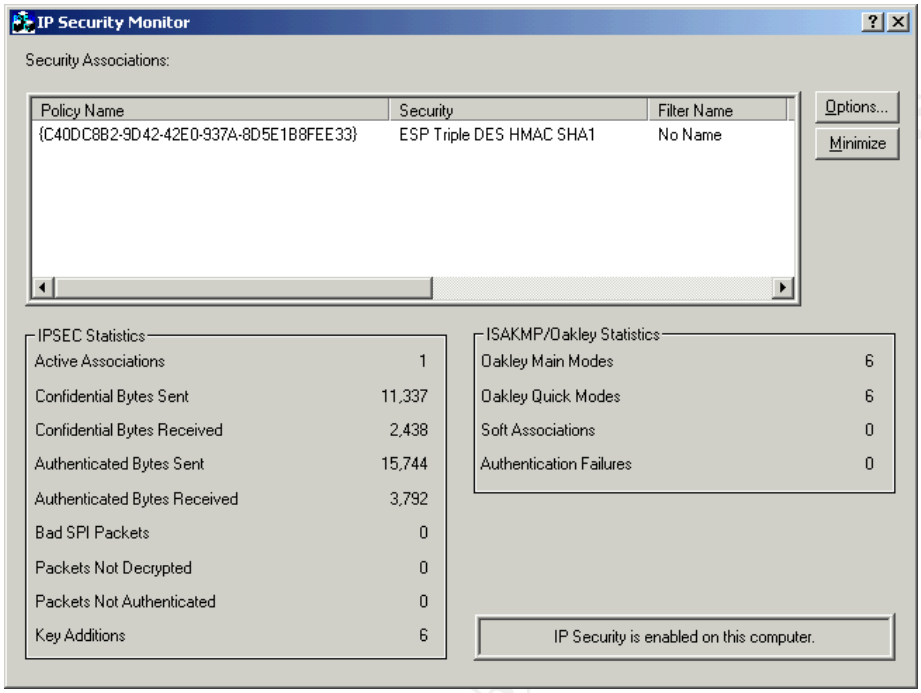


Right click on the policy and select "Assign". Repeat this on both peers. Try to ping the other peer and probably the following screen will appear. This is a good sign. :) Something is going on. :)



You can make sure what is going by executing ipsecmon. This utility will monitor the security

associations and show IPSEC statistics.



With this policy all traffic between the peers will be encrypted and there are some performance issues to consider. Depending on how much processor do you have available, you can install a network card with IPSEC “accelerator”. There are many vendors with card compatible with Windows 2000.

References:

- Microsoft's KB
- RFCs (1825-1829, 2401, etc)
- Windows 2000 Magazines
- SANS' "Windows 2000: IPSec, RRAS and VPNs" book