Global Information Assurance Certification Paper

Ben Freeman
Date Submitted: 11.21.2004
GCWN 5.0: Option 2 --Topics in Windows Security

Securing a Single Node Exchange Server 2003 Environment with OWA Support

Table of Contents

- Abstract 3
- Introduction 4

Exchange Server Features 4
- Client Support 4
- OWA 7
- S/MIME 8
- VPN 9
- Summary 9

Authentication 9
- Passwords and Passphrases 10
- Smart Cards 10
- SYSKEY 10
- Recommended GPO Settings 11

Encryption 13
- OWA 13
- SSL Implementation 14
- Forms Based Authentication Implementation 22
- RPC over HTTP Configuration under Exchange 2003 SP1 (Server Configuration) 24
- RPC over HTTP Configuration under Exchange 2003 SP1 (Client Config) 28
- Clients 32
- Intelligent Message Filter 33
- Domain Controllers 37
- IPSec 37

Other Considerations 38
- Anti-Virus 38
- Firewall 38
- Security Templates 38
- Patch Management 39
- Auditing 39
- Custom Scripts 40

Appendix A: Links 43

References 43

Abstract

This research paper will be geared towards the small to medium sized businesses that for whatever reason are not able to establish a Front-End/Back-End solution. I will begin by outlining the services provided by the Exchange Server itself so that the reader can get a grasp on the multitude of services to be potential risks. After a brief description of services to be provided I'll begin to assess what vulnerabilities are inherent in each and what the best course of action might be for preventing them. From there I'll begin outlining the best course of action for protecting the organization against potential vulnerabilities. I'll be covering best practices for Authentication, Encryption, Virus protection, Patch Management, and implementing Microsoft's new Exchange 2003 Plug-in called Intelligent Message Filter. IMF alone will reduce potential virus infections by leaps and bounds. I will also display a custom script that I made for detecting failed audits in the event log. This script will monitor the event log and send an email to an administrator the moment it detects one.

Introduction

Exchange Server 2003 is a full featured messaging solution for the Windows environment. In today's fast paced business environment E-Mail has quickly become common place in even the smallest of businesses for quick, efficient communication both in and outside the organization. While I'm sure internally controlled mail servers are a must for any large organization, for small to medium size businesses it isn't always necessary or cost efficient. Because of the initial costs and the maintenance associated with an in-house Exchange Server implementation, smaller businesses usually present the easiest targets for security breaches. They usually don't have the money or resources to devote to an ongoing secure Exchange or Windows Server environment. They often just need it up and running ASAP and it usually gets forgotten until something breaks. This is a hackers dream, and what this paper will attempt to help you prevent.

Exchange Server Features

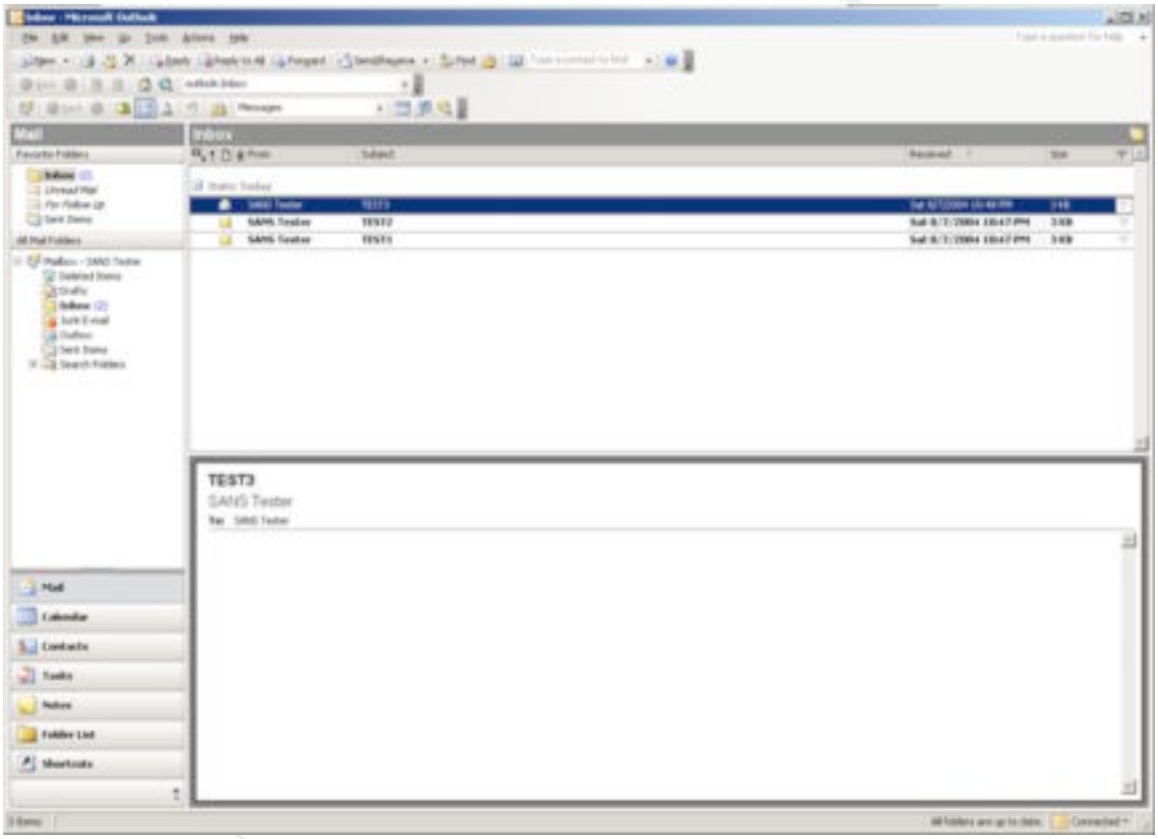
Client Support

There are a variety of different protocols available for accessing corporate mail through Exchange 2003. The three supported are POP3, IMAP4, and MAPI. For the purpose of this paper we will be focusing on MAPI supported clients, although I will give a brief description of IMAP4 and POP3 as well.
Post Office Protocol (POP3) is an older protocol that will work with just about any mail client on just about any operating system. Clients can choose to either leave mail on the server or remove it after downloading it. On the downside, that's about all the functionality you get with POP3. It's extremely limited and hence, very easy to setup.

Internet Message Application Protocol (IMAP4) is also supported by Exchange 2003 and is a little more complex than POP3 is. IMAP4 supports server side folders and storage of email folder hierarchies. It also allows server side searching of mail and the ability to download only particular messages to the client.

Messaging Application Programming Interface (MAPI) is the protocol we will be using primarily in our Exchange Server 2003 messaging environment. MAPI allows an extremely complex and feature rich messaging environment. Because MAPI is a Messaging API, it allows developers to develop solutions to interact with the Exchange Server in just about anyway possible. Some of the features available within Outlook 2003 when connecting to an Exchange Server are sending and receiving of mail, Public Folders, Tasks, Journal, Calendaring, Contacts, and Notes. All of this is information is saved on the mail server in the

form of a Mailbox. Other advanced features are also stored as part of this mailbox, such as Server Side Rules processing, and Free/Busy settings so that other users within the organization can check when you're in meetings and schedule resources correctly. One of the more difficult decisions to make is which clients should be allowed access to your messaging environment. There are a variety of ways in which you might allow employees access to your mail servers. The two main clients available for MAPI access include Outlook and Internet Explorer for Outlook Web Access. To complicate matters even more, there are a variety of different versions for each. We will be focusing on Outlook 2003 and generic web browsers as the clients. Below is a screenshot of the Outlook 2003 Client.



Most users will be using some version of Outlook (2000, XP, 2003) to access the mail servers within the organization. This is going to be the easiest and most secure form of access available to you. Users that are working within a Windows environment and only need access while at an employee site will almost always access the mail servers through the Outlook MAPI client. Through this client you will first have to be authenticated through a domain controller to gain access to the network, and as long as your clients are using Windows 2000 or later, will use NTLMv2* by default to do so. NTLMv2* is leaps and bounds more secure than previous forms of authentication used in Windows environments and is always the recommended method for authentication. Any client operating system using Windows 2000/XP/2003 will use NTLMv2*

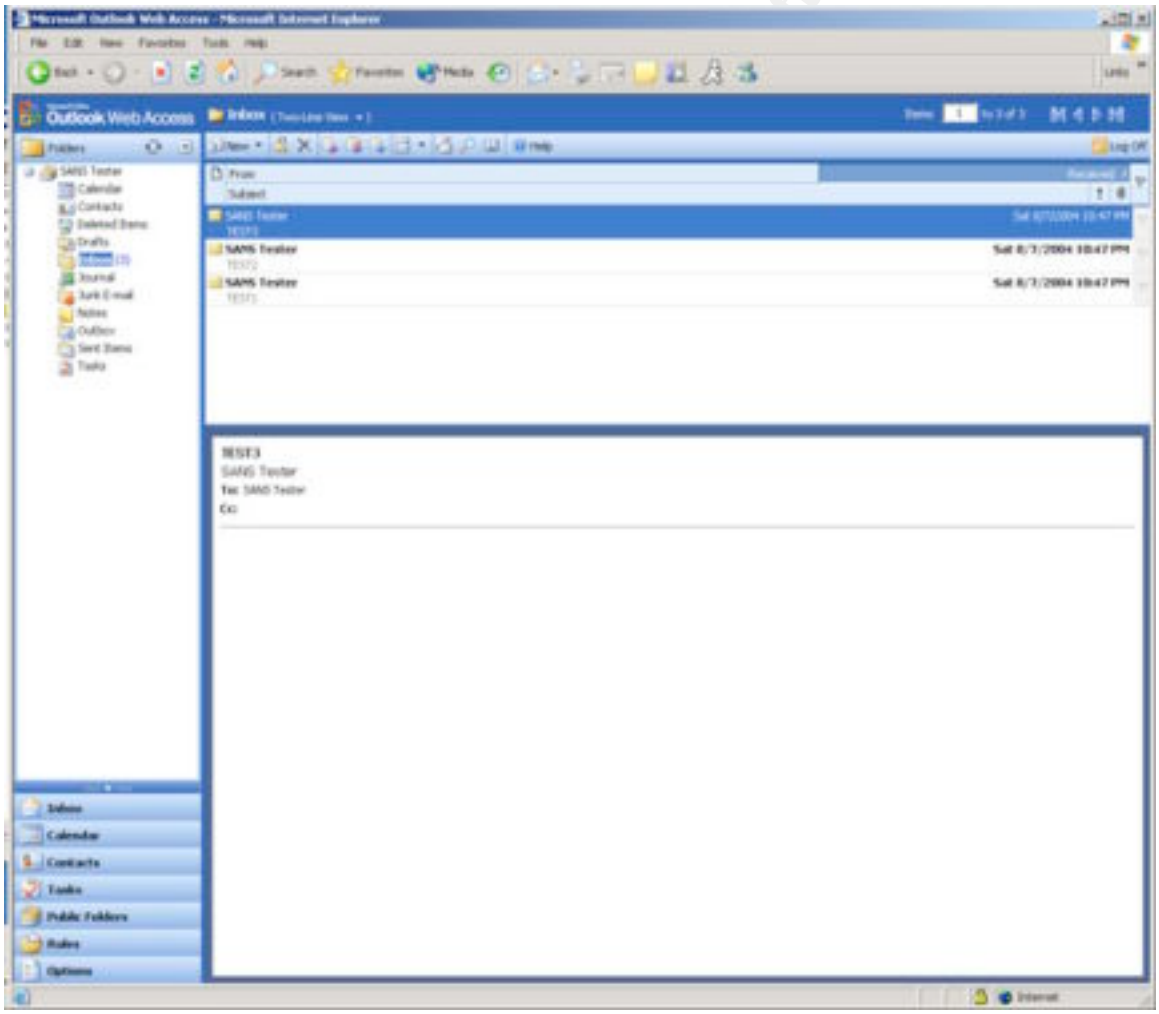
automatically, but you can also force it through a GPO. The only thing to consider before pushing through this setting is that some legacy applications that make use of authentication might not have NTLMv2* built-in functionality. As always, make sure to test everything in a testing environment before deploying to a production environment.

Another possibility for gaining access is through a web browser such as Internet Explorer, Mozilla, Firefox, or Opera. This is a great way to offer access to a variety of different client operation systems through a common interface. It's also a great alternative for telecommuters that for whatever reason don't have VPN access to the corporate network. The only problem with this method is that it allows for access from anywhere to a server behind your firewall through a web browser. You can imagine the plethora of different scenarios for access protection that should be in place if you are going to offer this service from the internet. Another thing to consider is that since just about any browser can access OWA, so can just about any client operating system. This means that you can not control the authentication protocol used by the client operating system. For any client connecting from a browser besides Internet Explorer you're going to have to enable Basic Authentication on the IIS web server to allow the client to authenticate. Passwords are sent in clear text by default with Basic Authentication, which means any hackers sniffing the wire can easily grab passwords with little or no effort at all. To prevent this from happening you're going to have to enable SSL on the web server accepting the OWA authentication requests. This means installing a certificate on the IIS web server and disallowing access to HTTP; only HTTPS access should be allowed.

The last form of client that could potentially access the mail server is a mobile device through what is called Outlook Mobile Access. Outlook Mobile access is beyond the scope of this document and will only be touched on. One thing I would like to point out about this form of access is the importance of determining which types of devices are accessing your corporate mail servers through what wireless gateways. There is a very important issue to consider and might be good enough reason just to disable this feature for now unless there is an extremely compelling reason your users need this feature. There is a known issue in the Wireless Access Protocol 1.x (WAP) gateways that leave your data unencrypted for a short period of time during transmission between the wireless gateway and your corporate mail server. When the data gets from the mobile device to the WAP gateway, there is a conversion in encryption that takes place. The WAP server decrypts the SSL and re-encrypts the data using Wireless Transport Layer Security (WTLS). During this conversion your data is left unencrypted for a short period of time. The fix for this is to either make sure that both the wireless device and the wireless devices' public gateway are WAP 2.x, or to host your own corporate WAP 2.x gateway.

OWA

Probably the biggest security threat of all for a Windows environment is running Internet Information Services (IIS). Most hackers will almost definitely get excited when they see an IIS server running in your organization. It's a favorite target for script kiddies and new security bulletins. Keeping your Exchange Server up to date with the latest security fixes is a **must** if you are going to make OWA available to your users. OWA runs on IIS and offers a very similar interface to your users as the Outlook client itself does. A lot of the advanced functionality of the Outlook client has been stripped out, but you can still access all of your mail, public folders, tasks, calendar and more. One exception is the Journal, which is still only available through the Outlook client itself. Below is a screenshot of what the OWA client looks like within Internet Explorer.



S/MIME

Secure/Multipurpose Internet Mail Extensions (S/MIME) is used to secure email within the organization. SMTP by itself was not meant as a secure method for transferring email between individuals and organizations. It was meant to be a very efficient protocol for mail delivery and not much else. Because there is no inherent security built into the SMTP protocol, every message sent with it is sent in plaintext across the internet for anyone sniffing the wire to read. S/MIME attempts to solve this problem, and with its use becoming more and more widespread might just do so. The current version of S/MIME is version 3, and is supported by more and more clients everyday. S/MIME achieves higher levels of security with the SMTP protocol by enabling two main services, Digital Signatures and Message Encryption.

The Digital Signature part of S/MIME provides three key security benefits for sending email. Authentication, nonrepudiation, and data integrity are the key features provided by these. A signature can be thought of like a legal signature on a document, from the signature itself you know who signed it. When a user sends an email they attach their signature to the document as a form of authentication so that the receiving party knows who the document came from. Normally with SMTP alone there are no authentication mechanisms, anyone can send a message saying they're whoever they want. With signatures you know who the message came from by the signature attached to the message. This also supplies nonrepudiation for important binding documents. The fact that the email has been signed by a party means that it can be proven it came from that source. This is very useful for proving a message came from somewhere when someone is denying sending such a message. With SMTP, a user can easily say they never sent a particular message and it becomes particularly difficult to prove them wrong because of the lack of built-in authentication. The last security feature provided by S/MIME is data integrity. This means that an email can not be altered in transit. If the message is altered by a third party in transit the signature on the document does not match up on the receiving end. These three features of added security alone make S/MIME an extremely important service for securing email on the internet.

The second major security feature S/MIME provides is message encryption. Message encryption provides confidentiality for both parties while the message is both in transit and in storage at the receiving end. When a normal SMTP message is sent across the internet it is not encrypted at all and is sent in plaintext for anyone who wishes to read. With message encryption the messages are encrypted when being sent, sent over the internet encrypted, and decrypted at the receiving end. This does not provide any sort of authentication or nonrepudiation since it's only securing the contents of the message itself, not authenticating the sending user. Message encryption does in fact provide data integrity though since the message can not be read in transit, there is no way for someone to alter its contents.

All in all the combination of the two services is what provides for the enhanced security of the SMTP protocol. If only one or the other is used only certain aspects become inherently more secure. If only encryption is used then

only message privacy is achieved, but you still have no way to prove where or who is was sent from. If only digital certificates are used than you can prove where the message came from and who sent it, but not whether or not the contents of the message itself has been read by third parties, or altered in anyway in transit or after being received. The combination of the two services is an essential part of email security going forward.
-- A lot of the information contained in this S/MIME section was gathered from the "Exchange Server 2003 Message Security Guide[4]"

VPN

One major consideration for access to your Exchange Server or any corporate resource is how you will allow external access. For the most part you have three main options for external access. The first is OWA, which was described earlier n the OWA section. The second two methods for remote connectivity are Virtual Private Networks (VPN) and Remote Procedure Call (RPC) over HTTP. VPN is currently the most common implementation for all external access to corporate resources. The protocol forms an encrypted "tunnel" over the internet to your corporate network. Because the internet is used as the primary vehicle instead of expensive leased private lines, it's a very cost effective alternative for most companies.

RPC over HTTP is a new feature that was added with Exchange Server 2003. It's pretty complex to setup but with the added support for a single server implementation with Exchange Service Pack 1, it makes it a great alternative to supporting a VPN infrastructure. I will be detailing the setup later in this paper. The concept is that RPC packets are encapsulated and sent over HTTP port 443 to enable VPNless connections to corporate mail servers. Because all data is sent via HTTPS, the packets are encrypted end to end.

Summary

Now that I've gone through all the various situations and services that could arise within an Exchange Server 2003 environment we will begin to assess risk and explain how to mitigate such risks. Nothing is perfectly secure, possibly nothing ever will be. There are new exploits daily and new code to exploit being produced daily. The best security is active management and monitoring of your servers and software applications.

Authentication

Passwords and Passphrases

Passwords as we know them today will never be able to satisfy the needs of a high security environment. They are just too easily crackable by anyone with enough free computing cycles. Passphrases, on the other hand are easier to remember and much, much harder to crack. Consider the fact that the amount of possibilities for the hacker to crack goes up exponentially with each letter added to the passphrase. The sentence “1 have a cat named Baby, she is a cutie!” has 39 characters in it! This would never be cracked by anyone besides maybe the NSA, and probably only then if you were a serious government threat. On the other hand, the traditional 6-12 alphanumeric password with numbers, capitals, and symbols such as “A1bw&LK9” is leaps and bounds easier to crack and harder to remember. I guarantee you will find this password on a sticky next to the monitor, whereas anyone can remember a sentence about their cat.

Smart Cards

In addition to a strong passphrase based security model, smart cards should be used if possible. Smart cards will quickly become more and more popular as security becomes more of an issue for companies. It’s not a free option like passphrases are so most companies won’t invest in them unless they can see evidence of hackers affecting their bottom line. More and more computers are coming standard with smart card readers though, and the setup and configuration isn’t exactly mind boggling. A smart card is a tiny credit card size device that can store information about a user such as certificates and passwords/passphrases. It handles all the cryptographic operations instead of the operating system and is used instead of typing in a username and password for authentication. This makes logging onto any computer physically without the card itself impossible. The card is protected by a PIN, which will disable itself after a certain amount of bad attempts to prevent stolen cards being used. There is a GPO to force smart card authentication for any AD account, and one to lock a workstation the moment one is removed from the computer. I highly recommend these if you can slip them into the budget at all.

SYSKEY

SYSKEY is a utility that was put out to encrypt password hashes in the SAM database to defend against L0phtCrack and other brute forcers. The encryption it provides is turned on by default in Windows 2000/XP/2003 and can’t be disabled at all. There are a few options to configure though. SYSKEY will protect your data by creating a 128-bit RC4 “System Key” and this key can be stored in a number of different ways. As you can see in image 2.x, there are a number of different ways to store the system key. I am going to recommend that you use the floppy drive method for any install with a decently high physical security. With the floppy disk method you can make multiple backup copies of the disk in case one gets corrupted or stolen. This can also provide unattended reboots to occur. With the password at startup option, someone must physically

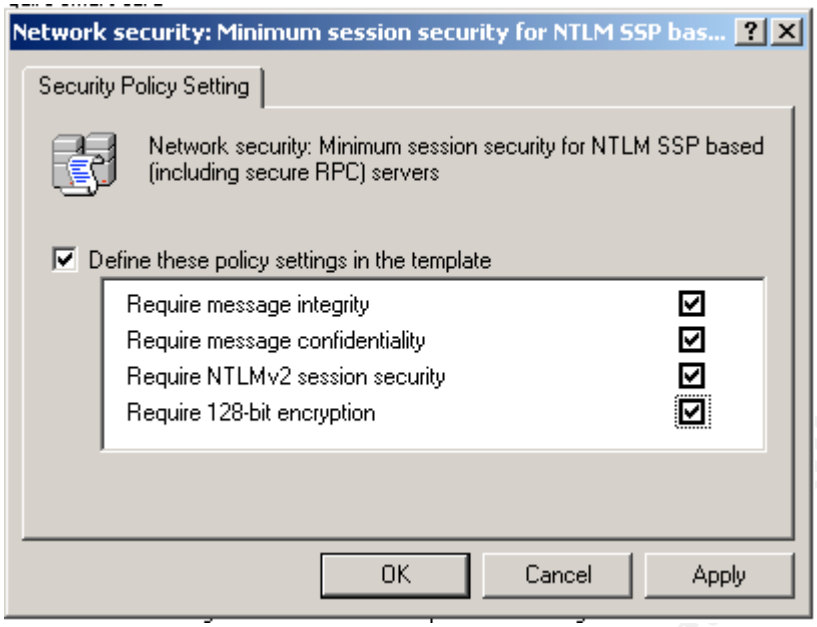
be at the machine during boot, which is oftentimes not possibly for servers in data centers or at other remote locations. The third option store the key locally isn't a bad choice either, since it's not required you be at the server and type a password. It also offers the same protection the other methods do. The only problem with this method is that the key is stored in an undocumented place in the registry, but "obscured" in a way that no one is able to find it. Problem is that lots of people are looking and who knows when someone will find it! Suggestions about the usage of SYSKEY were taken from "Windows 2000/XP/2003 Active Directory[2]"

SYSKEY.EXE

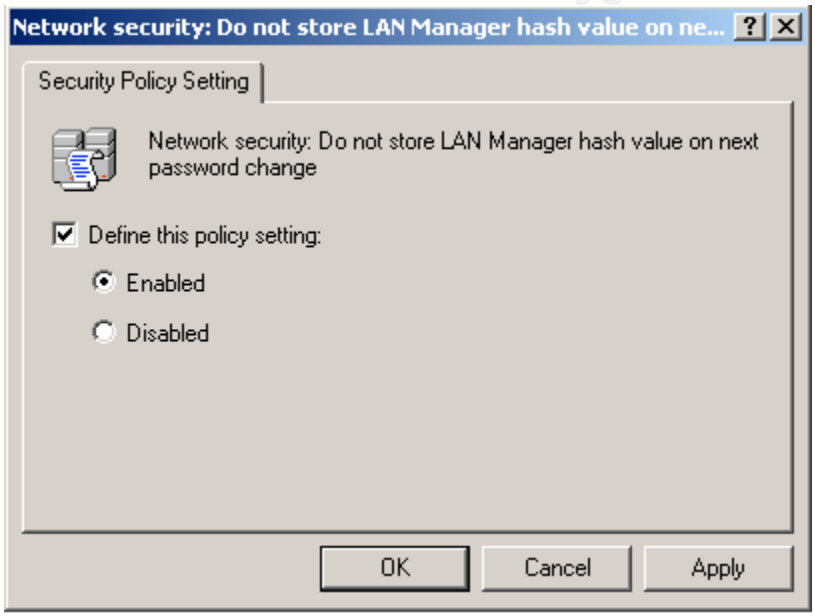
Recommended GPO Settings

Below are the recommended GPO settings for enhanced password and encryption between clients and servers. As always, legacy applications might not work with advanced security features. Because of this, any adjustments made should always be first tested in a replica test environment before being introduced into a production environment. All the recommended GPO settings were taken from ""Windows 2000/XP/2003 Active Directory[2]"

Enable these at the Default Domain GPO.

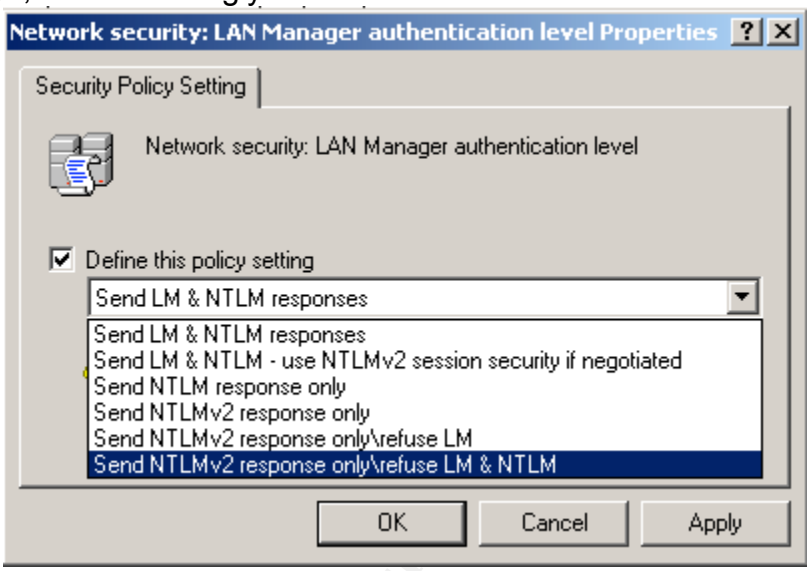


Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options



Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options

- Enabling this GPO will make sure that legacy authentication methods are never used. This will prevent older security threats and bad encryption to be used on your network. Anything less than NTLMv2 is a pretty large threat to your entire domain and should not be allowed. Your only consideration with this setting should be legacy application support. As always, test accordingly.



Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options

Encryption

OWA

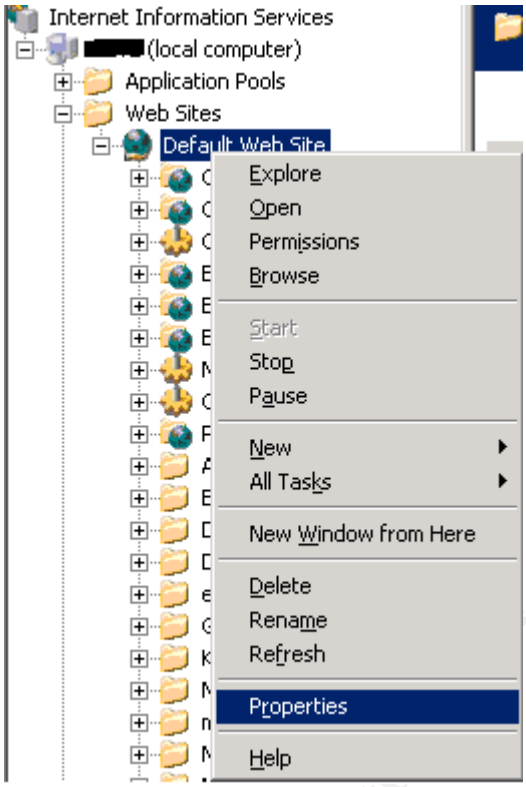
For OWA there are two considerations to keep in mind. One is the fact that this means you're running IIS on the Exchange Server. Locking down IIS properly to protect it from hackers is outside the scope of this document and is a massive topic. One important defense IIS 6.0 has implemented by default is the IIS Lockdown tool. With Windows Server 2003 this has been imbedded into IIS and will help protect against a large number of known vulnerabilities straight out of the box. Another great tool to make sure you use is URLScan. This is installed with IIS 6.0 by default but not completely configured. It is highly recommended you configure this tool to lockdown the commands that are able to be issued to your Exchange Server. There are recommended settings for this tool depending on how your Exchange environment is setup. This table can be located in the "Exchange Server 2003 Security Hardening Guide[3]".

Probably the most important security feature that's should be enabled for OWA is SSL. There is no reason access to OWA should be allowed over HTTP. There are actually two security features that should be enabled via SSL. The first is HTTP over SSL, and the second is Forms Based Authentication. To get SSL working you first need to install a certificate on the Exchange server and then

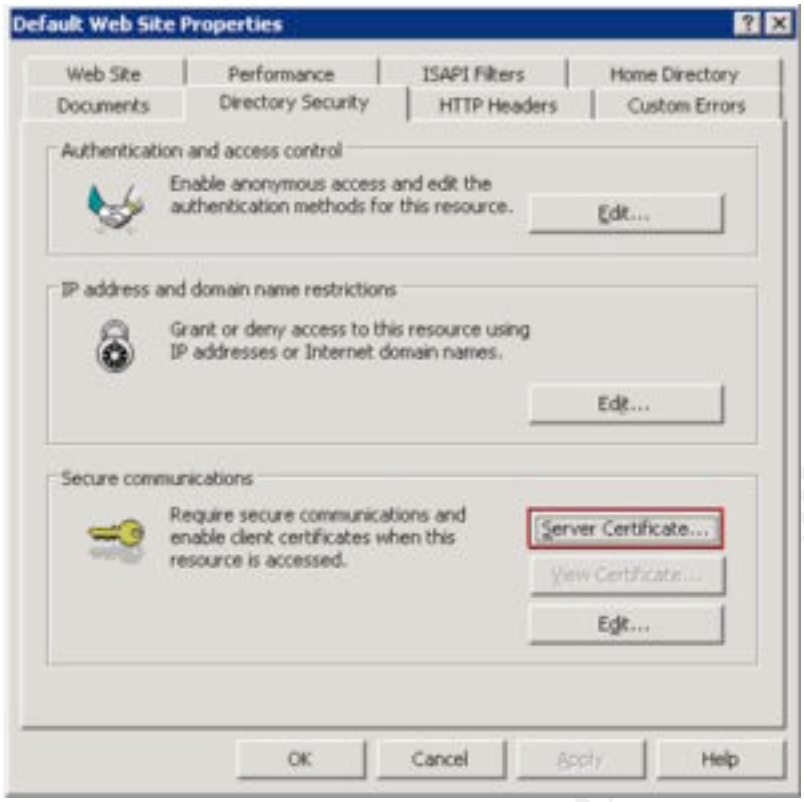
enable it within IIS. Below are the steps for getting the server certificate, SSL and Forms Based Authentication installed and working.

SSL Implementation

-- Steps for installing the web server cert with Enterprise CA already setup.

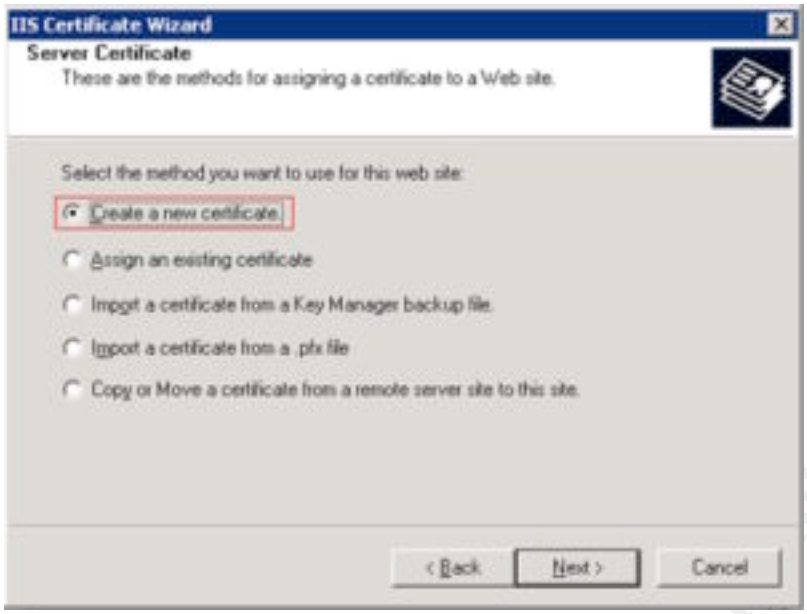


Open IIS >
Right click default Web Site >
Properties

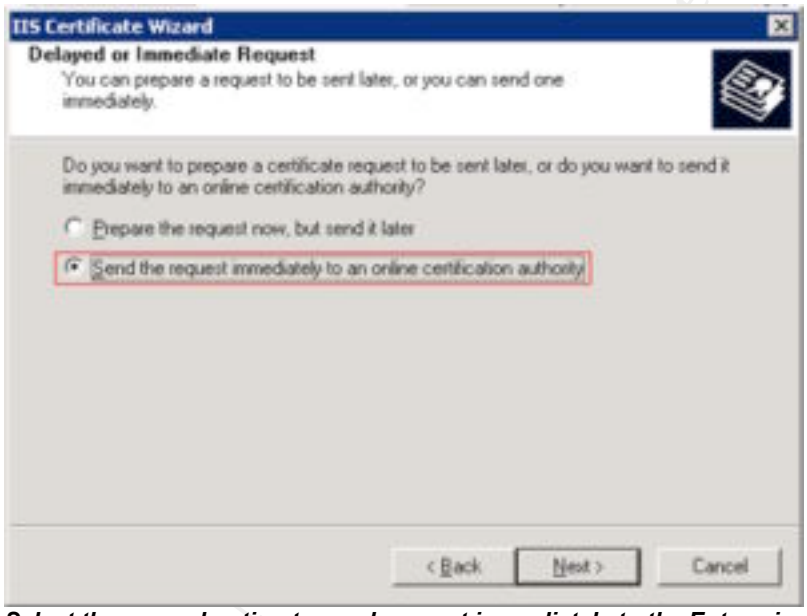


Click "Server Certificate"

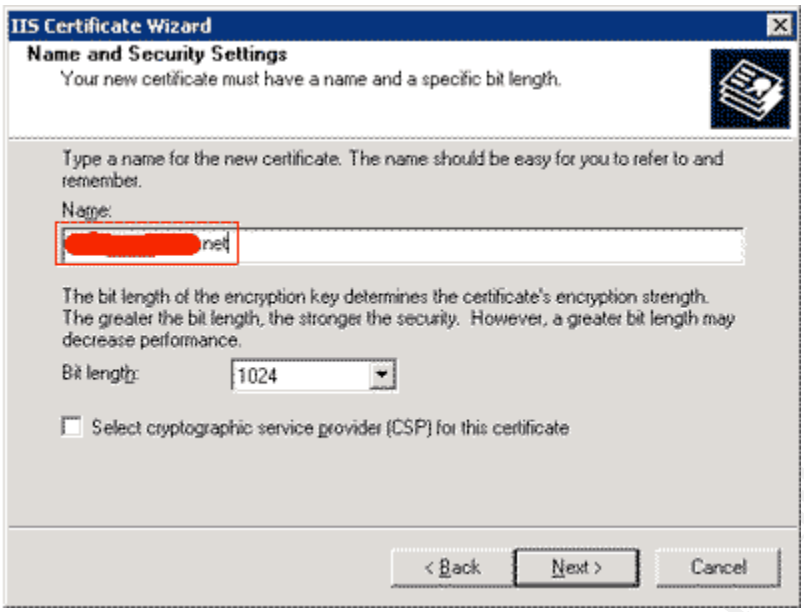
Click "Next"



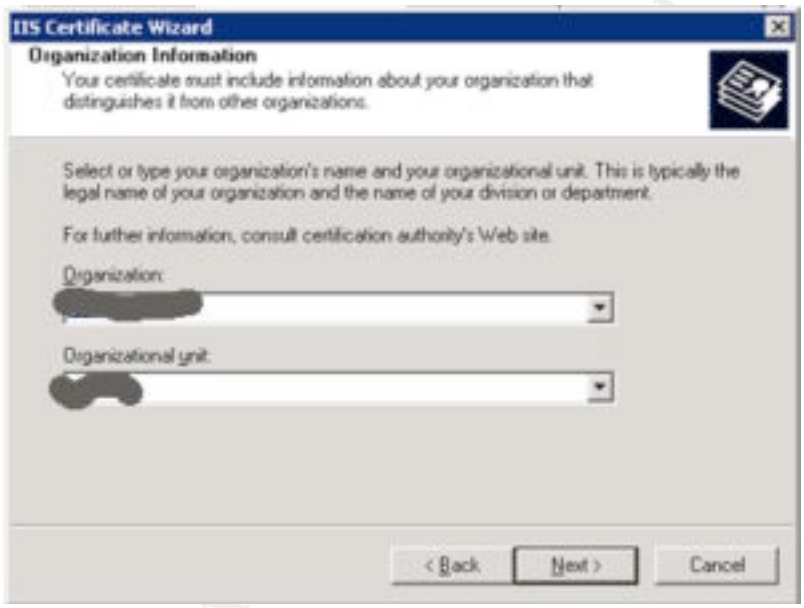
"Create a new Certificate" then click "Next"



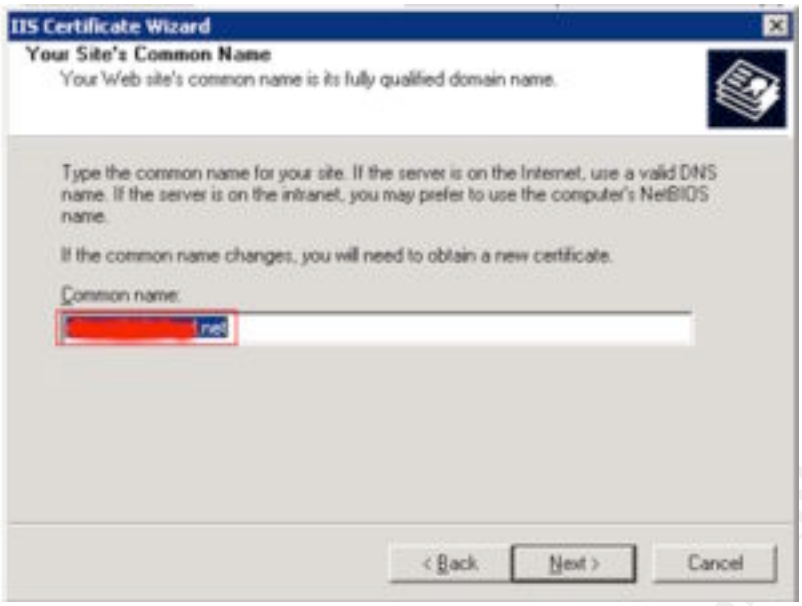
Select the second option to send request immediately to the Enterprise CA and then click "Next"



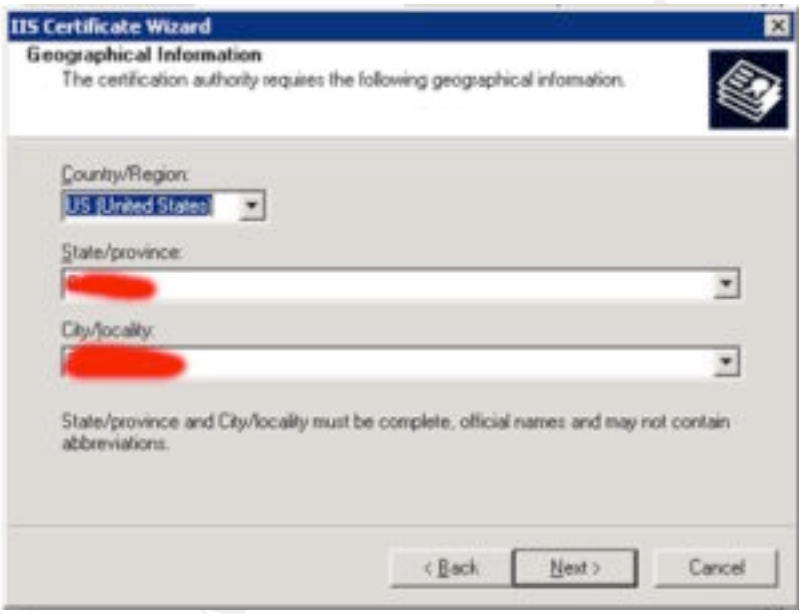
Make sure to put the FQDN for the web address users will enter when going to the site then click "Next"



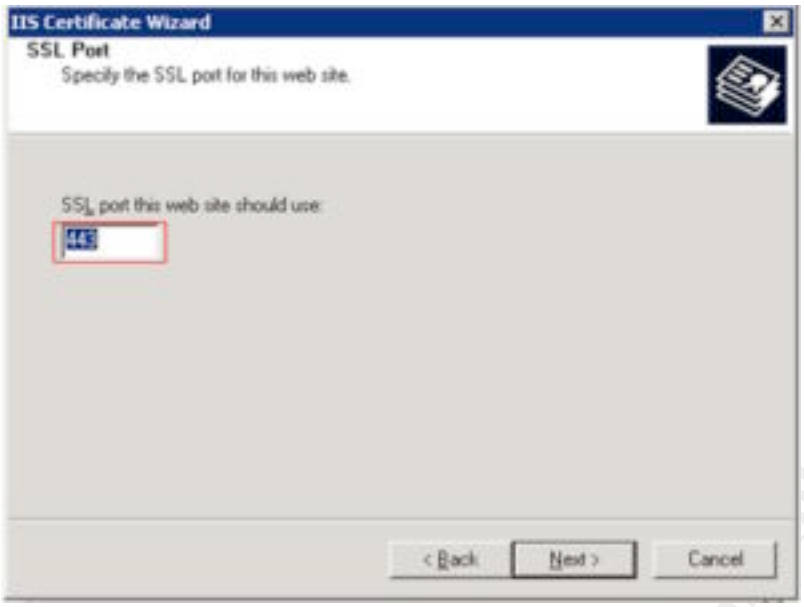
These two fields don't affect functionality at all and are only used for identification purposes when others view the cert. Click "Next" when done.



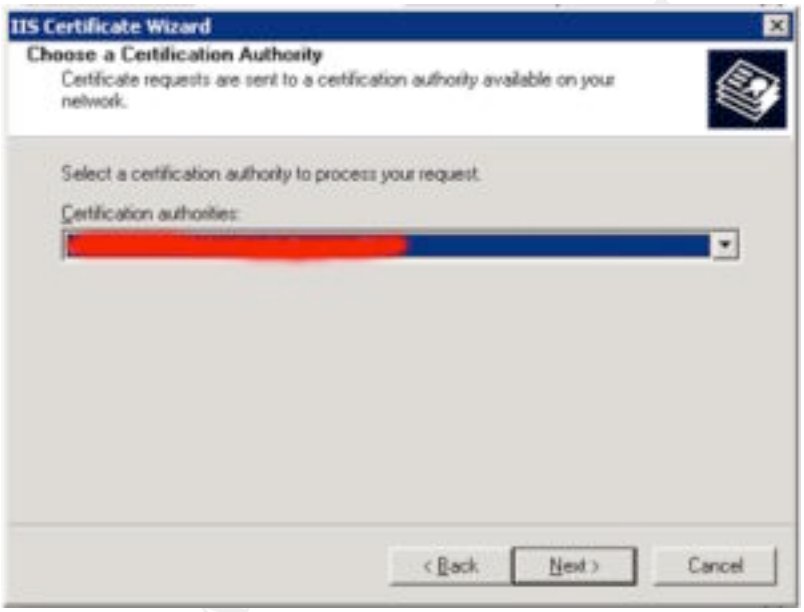
Again, enter the FQDN that users will use for the web address when accessing OWA and then click "Next"



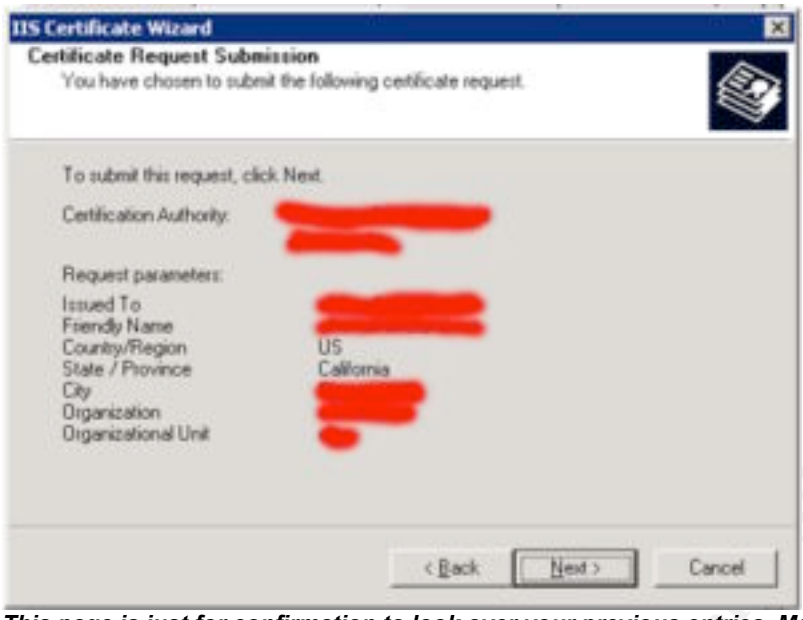
This is again just for identification when others view the cert. Click "Next" when done.



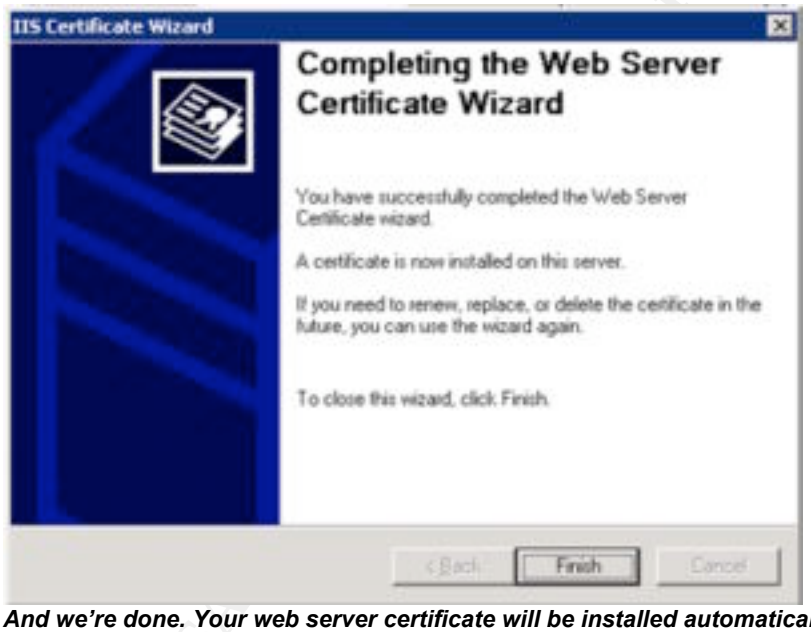
To reduce headaches, leave the default SSL port at 443. Not all applications even allow you to change this port and it is the standard for SSL. Click "Next"



Here just make sure you have the Enterprise CA within your domain selected. This will be the server to issue your cert. Click "Next"

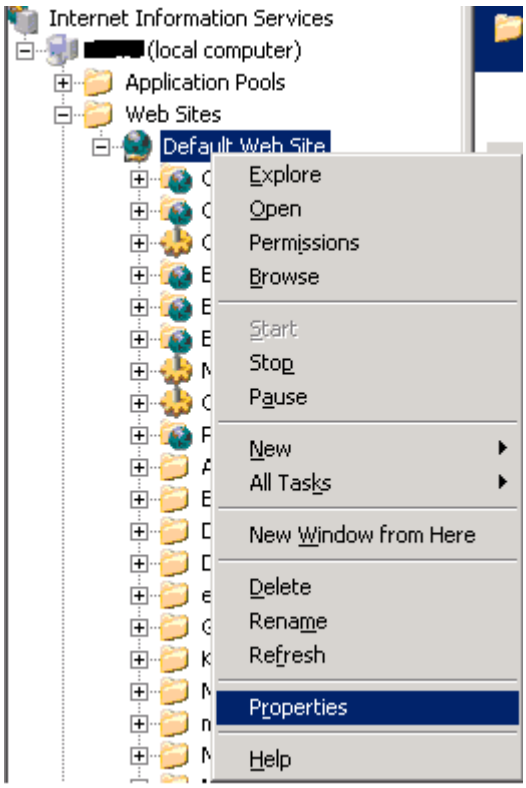


This page is just for confirmation to look over your previous entries. Make sure everything looks okay and then click "Next" and it will submit the request to the CA selected in the previous step.

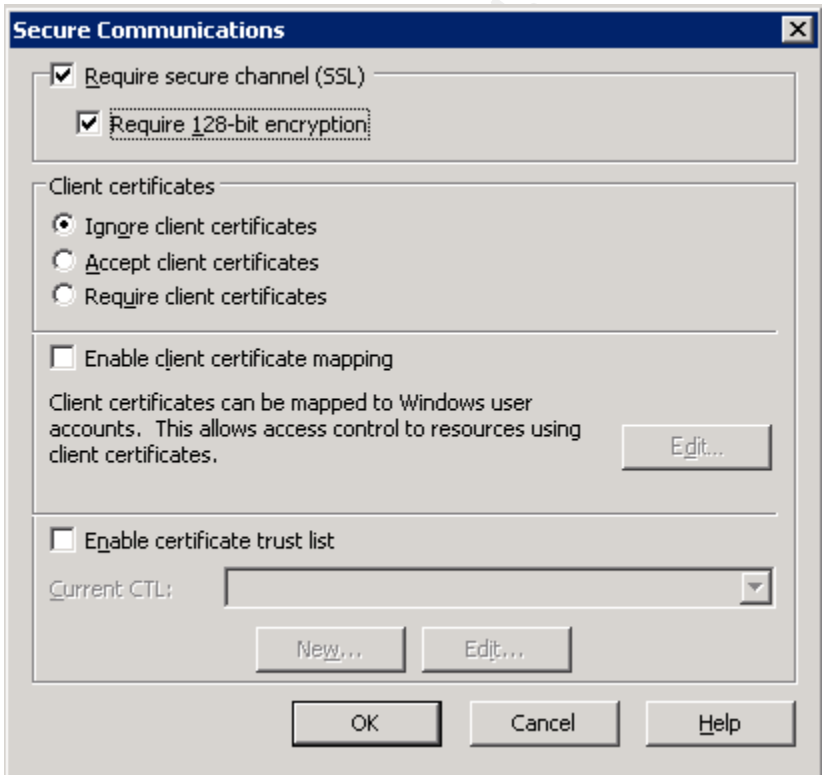


And we're done. Your web server certificate will be installed automatically and you can now enable SSL.

-- Enabling SSL on the Web Server



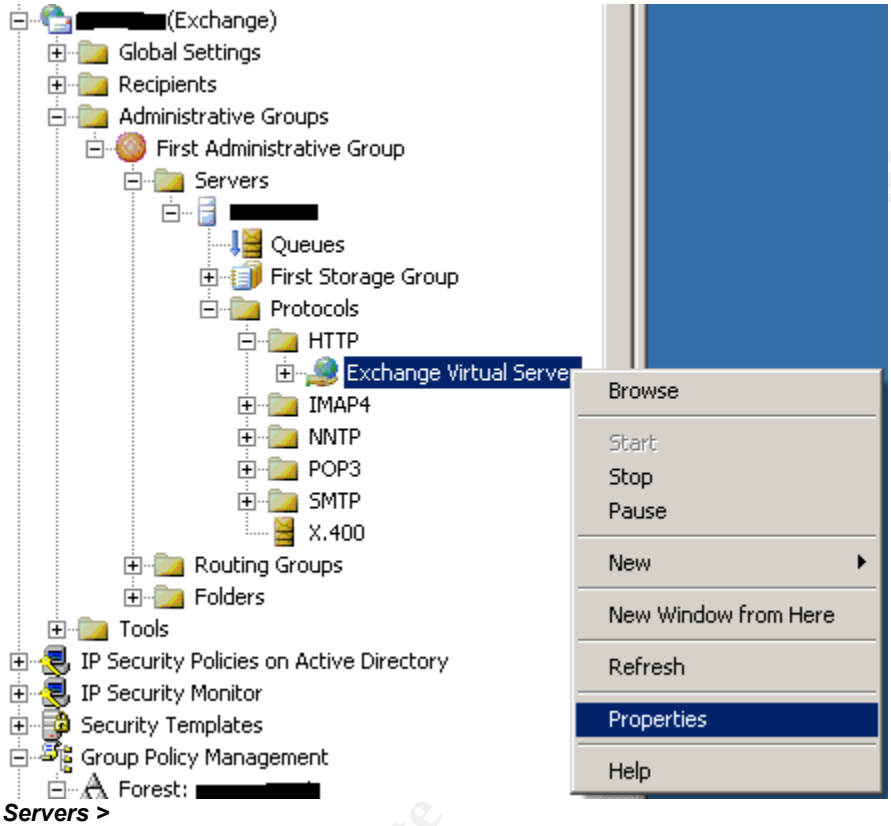
Open IIS >
Right click default Web Site >
Properties



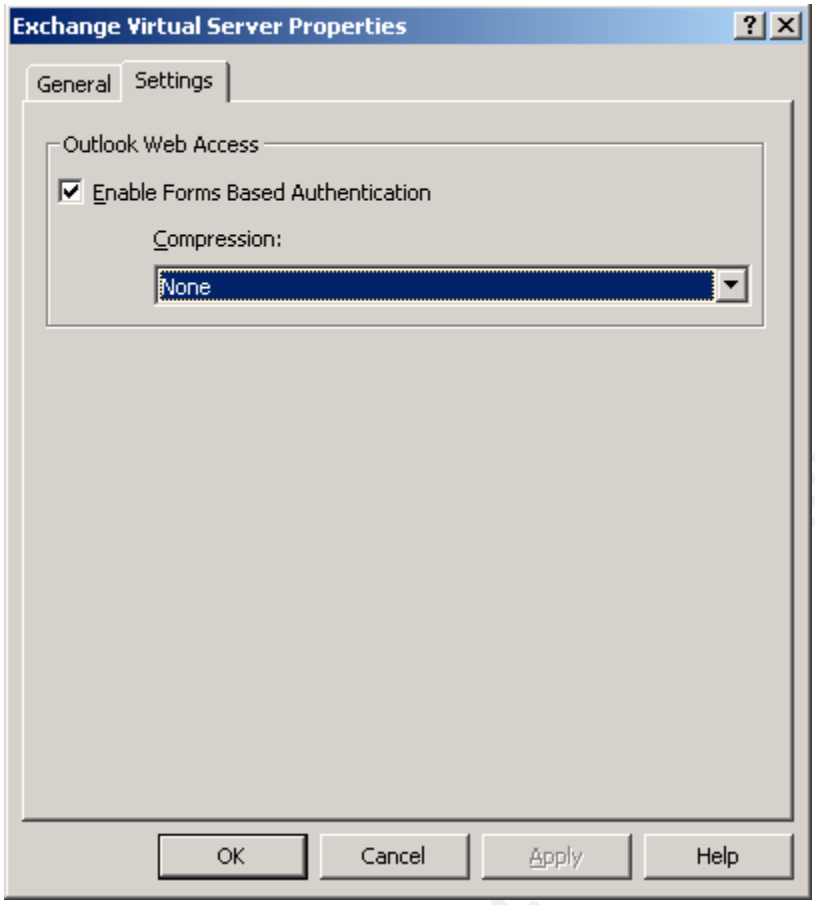
Directory Security Tab >
Under "Secure Communications" select EDIT >

Make sure the two checkboxes shown above are selected

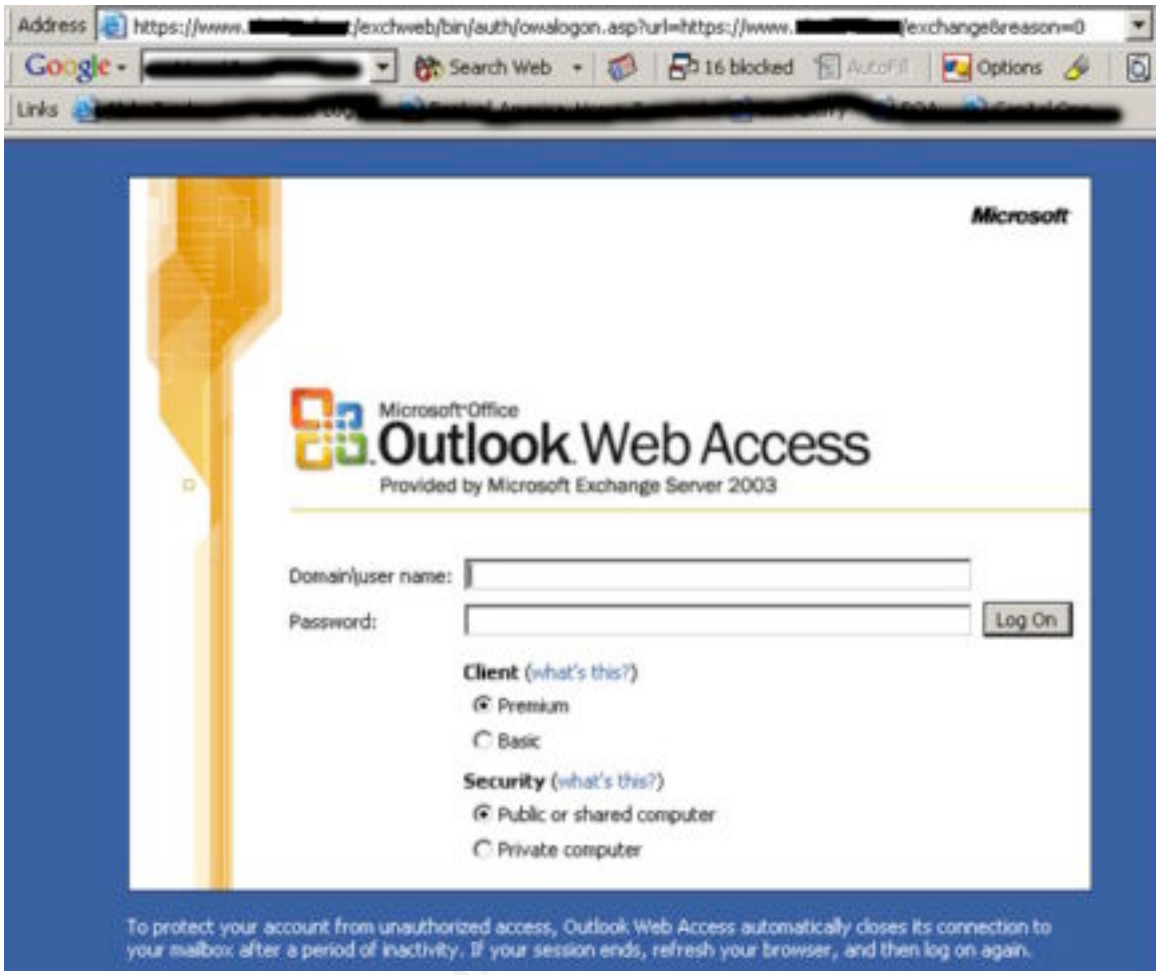
Forms Based Authentication Implementation



Servers > OwaServerName > Protocols > HTTP > Exchange Virtual Server

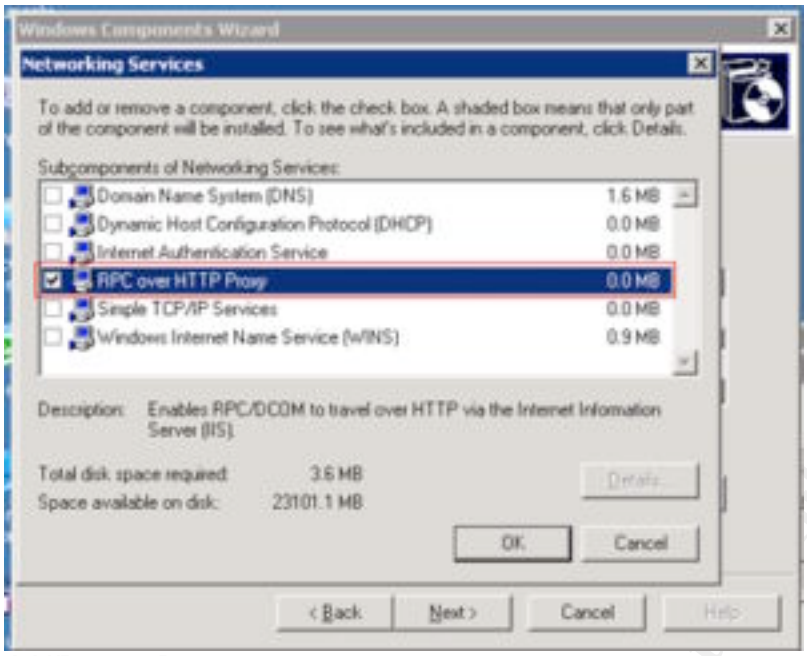


Make sure the checkbox above is selected to enable FBA



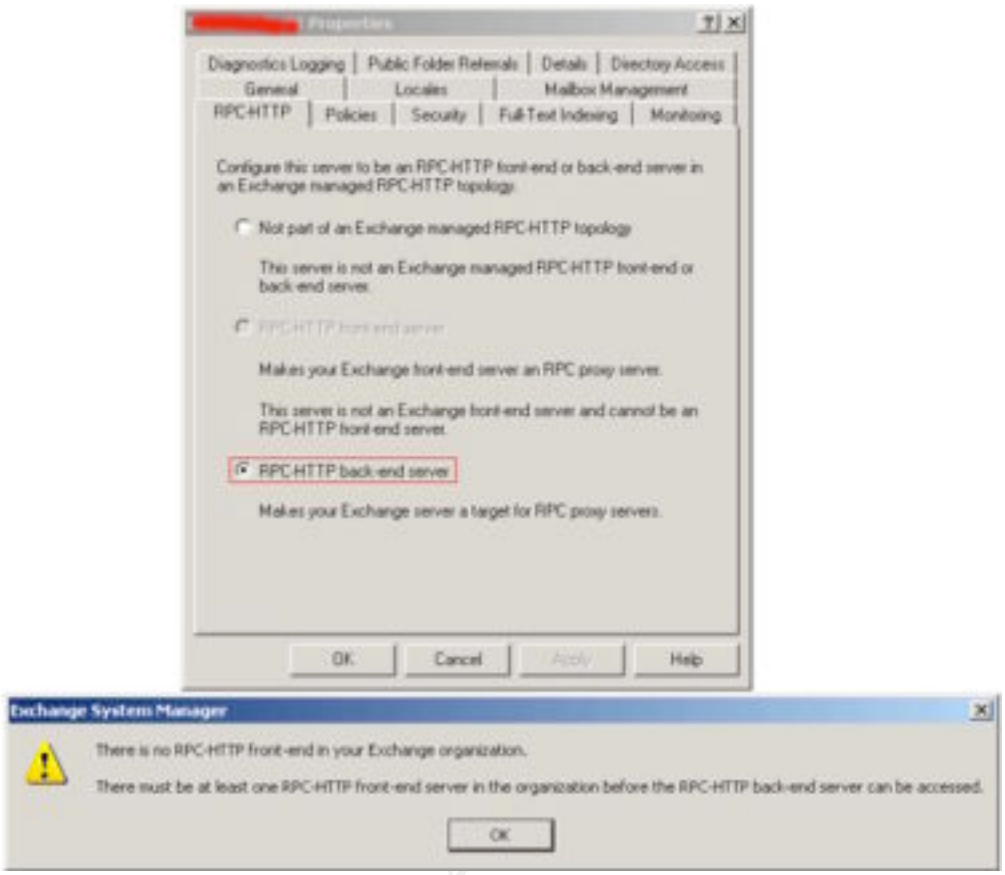
This will be what the Forms Based Authentication will look like. HTTP will no longer work and only HTTPS will be allowed.

RPC over HTTP Configuration under Exchange 2003 SP1 (Server Configuration)



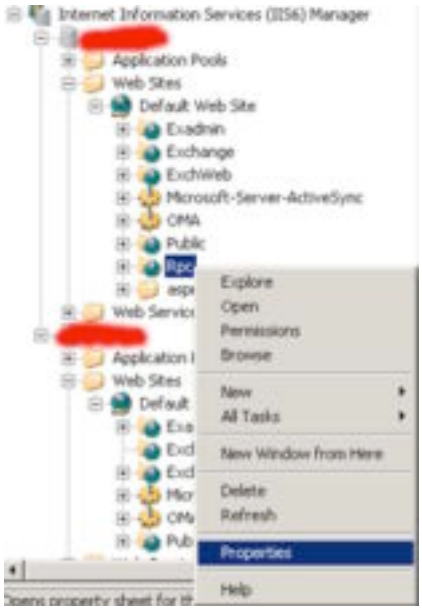
Add/Remove Programs >
Windows Components >
Networking Services >
RPC over HTTP Proxy

After installing that new networking component you'll notice a new Virtual Directory within IIS called "Rpc". Before we get to that though, go into the ESM (Exchange System Manager) for the next step below.

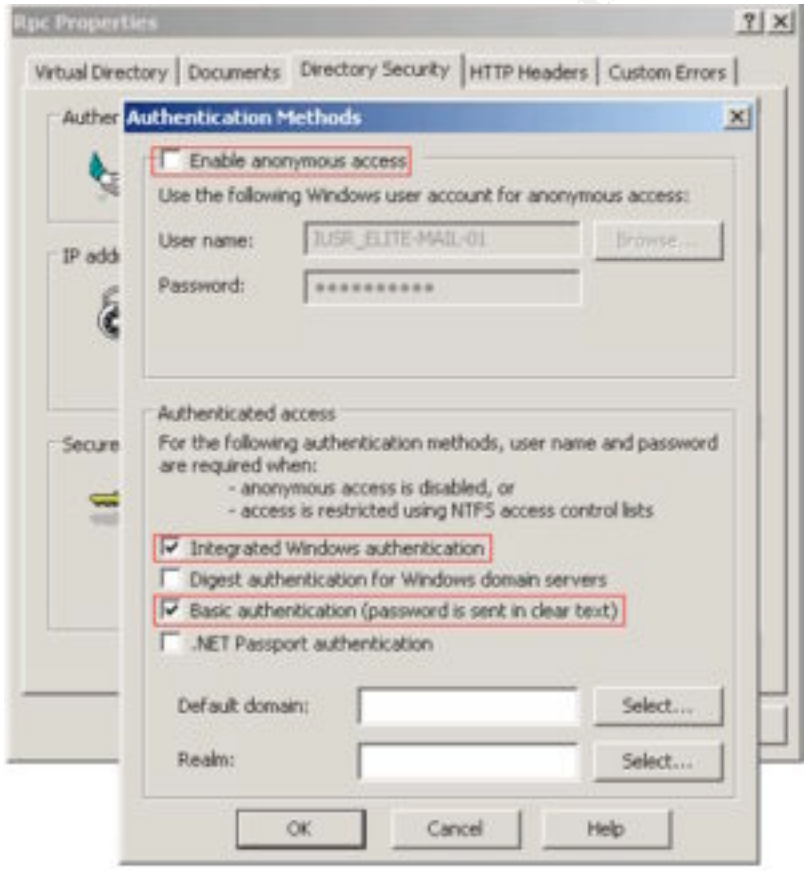


Expand ESM Admin Groups till you get to your Exch2003 Server.
Right click the Exchange Server then go to "Properties".
Next click the RPC-HTTP Tab and finally click the RPC-HTTP back end server radio button as shown highlighted above.
You'll notice the message below it, this can be safely ignored.

Next we need to configure the new RPC Virtual Directory within IIS.



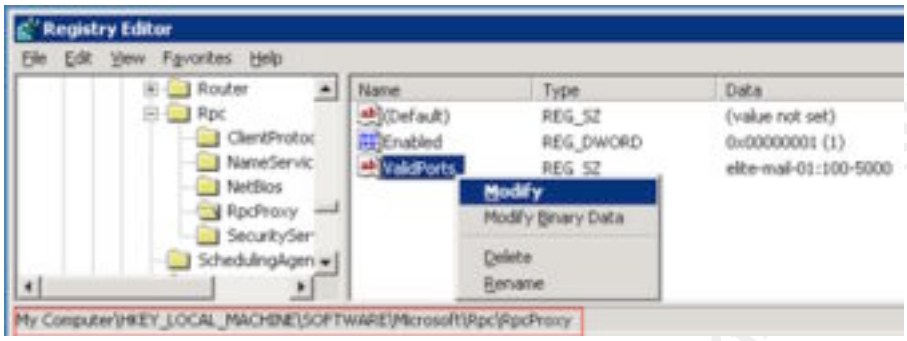
IIS MMC >
ServerName > Web Sites >
Default Web Site >
Right click Rpc >
Properties



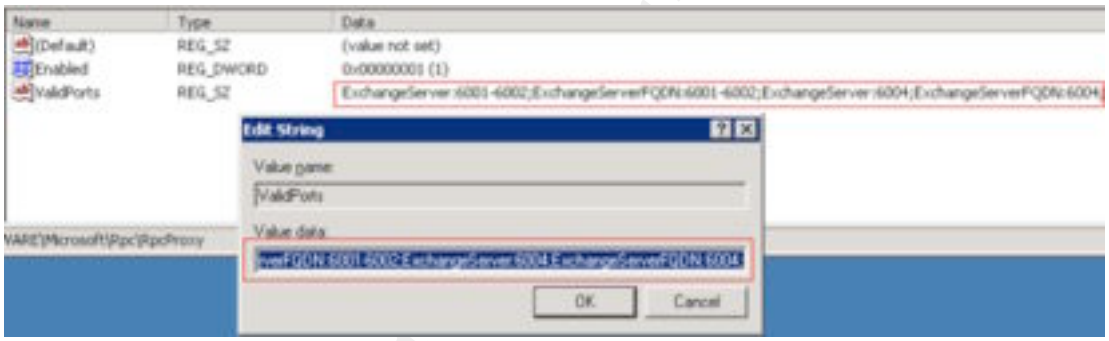
Click the Directory Security Tab >
Click the "Edit" next to Authentication and access control >

Make sure "Enable anonymous access" is unchecked >
Make sure "Integrated Windows Authentication" and "Basic Authentication" are checked

Next we must configure the Exchange Server to use certain ports for RPC over HTTP. To do this you must edit the registry as shown below. This is necessary because normally RPC ports are chosen at random.



Open Registry Editor on the Exchange Server (RPC HTTP Proxy Server) >
Browse to this key: HKEY_LOCAL_MACHINE\Software\Microsoft\Rpc\RpcProxy >
Right click the "ValidPorts" subkey and choose "Modify"



Within the "Value data:" text field enter the following below.
ExchangeServer:6001-6002;ExchangeServerFQDN:6001-6002;ExchangeServer:6004;ExchangeServerFQDN:6004;

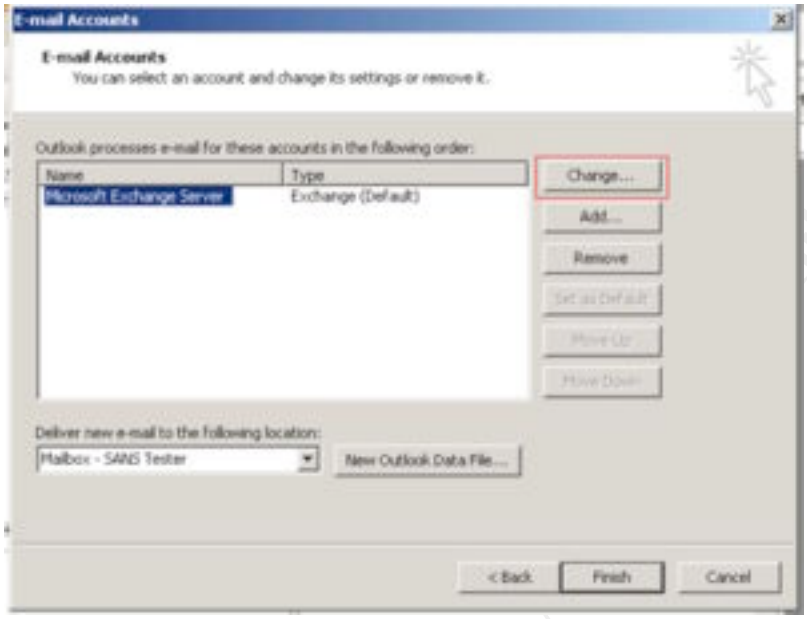
Keep in mind two things when entering that string:
"ExchangeServer" should be the Netbios name of your Exchange Server.
"ExchangeServerFQDN" should be the FQDN of your Exchange Server.
Change those two according to your Exchange Organization.

RPC over HTTP Configuration under Exchange 2003 SP1 (Client Config)

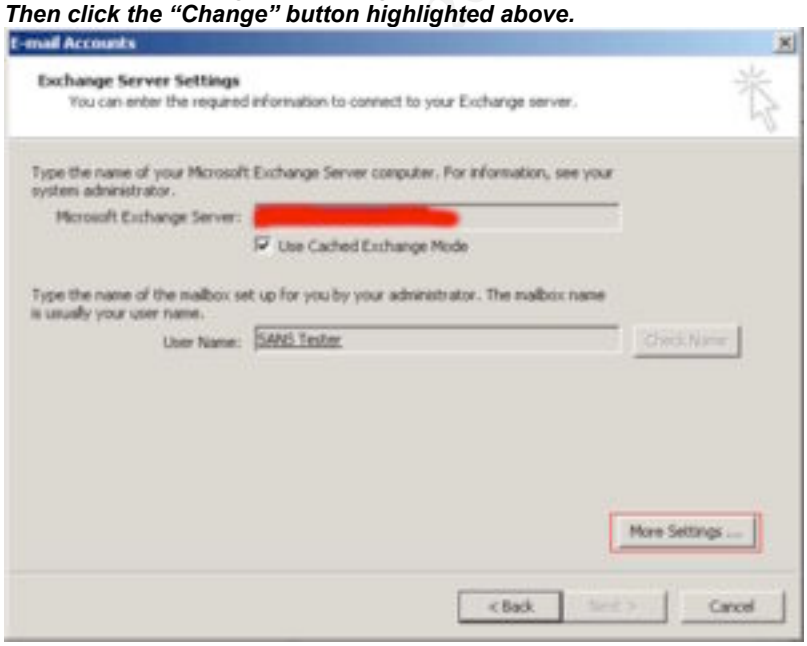
For client configuration Outlook 2003 is required and it's recommended you install Windows Service Pack 2 and/or install this update: Hotfix Q331320.
Another thing to keep in mind is that it's required you be connected either via VPN or directly into the corporate network during initial setup of HTTP over RPC. I have heard reports of the initial setup working without this requirement, but have not been able to duplicate it.

Firewall Settings: Open Port 443 and 25 and nothing else for maximum security.

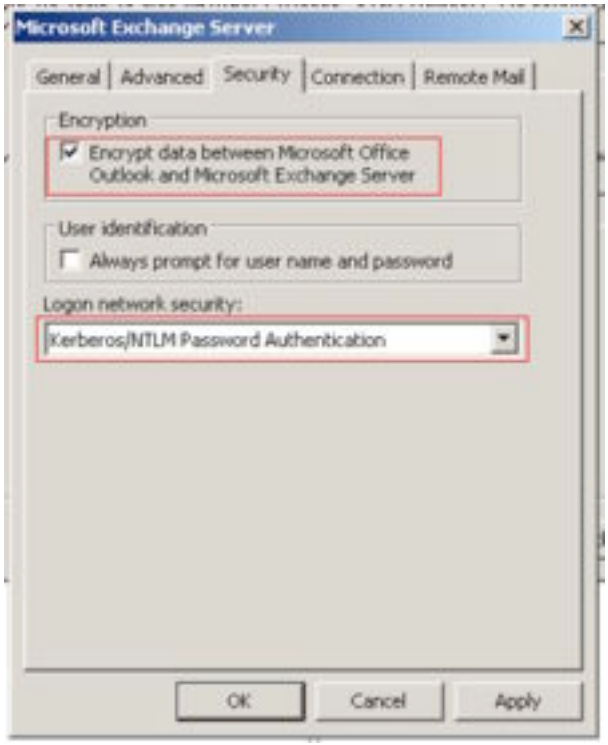
Now I'll go over configuring the client to access the Exchange Server via HTTP. To start go into the account properties for the Mailbox you wish to enable.



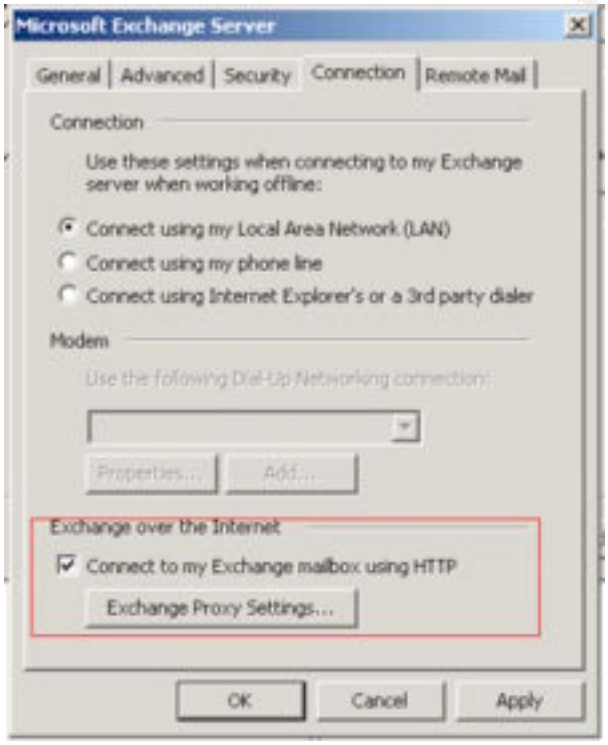
Within Outlook:
Tools >
E-Mail Accounts >
Click "Next" to change an existing account. >
Then click the "Change" button highlighted above.



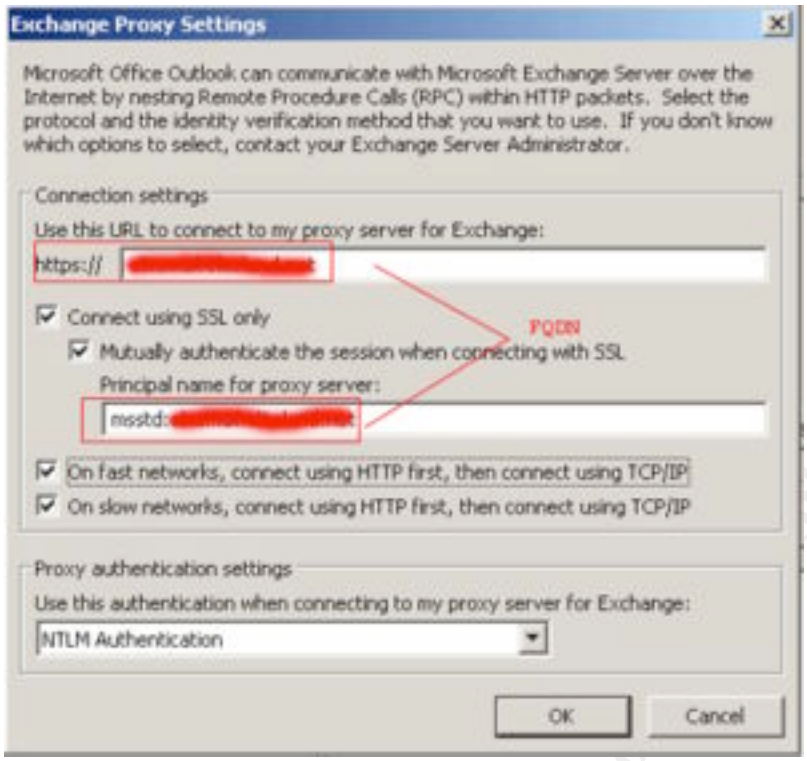
Next click the "More Settings" button



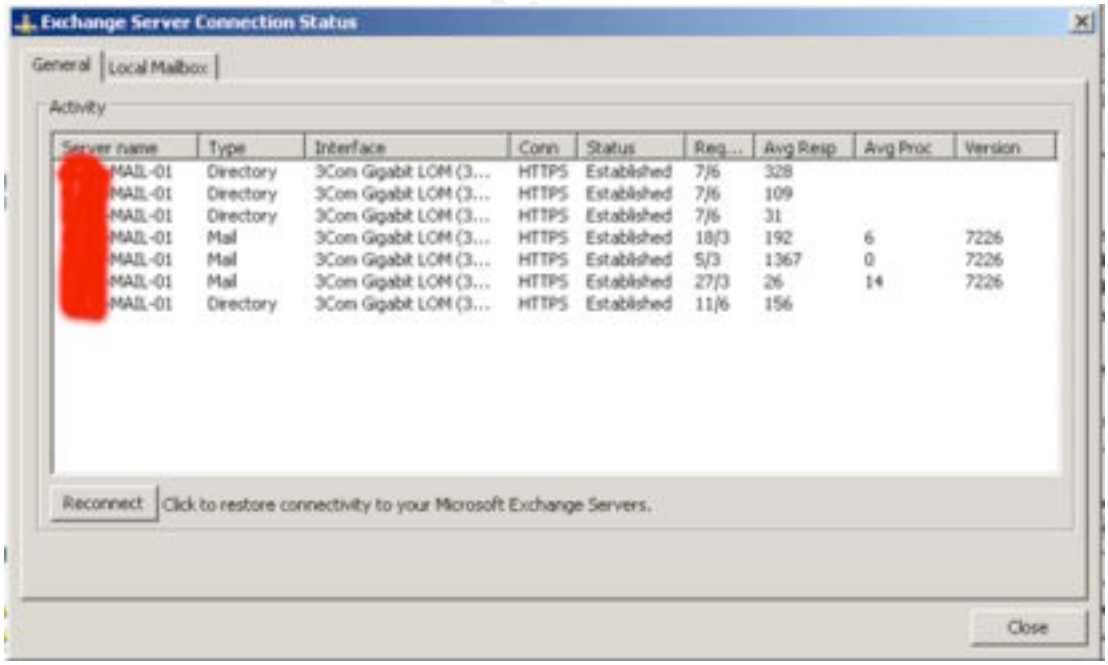
Make sure the Encryption box is checked and that logon security is set to the above setting.



On the connection tab make sure the box shown above is checked and then click on the "Exchange Proxy Settings" button.



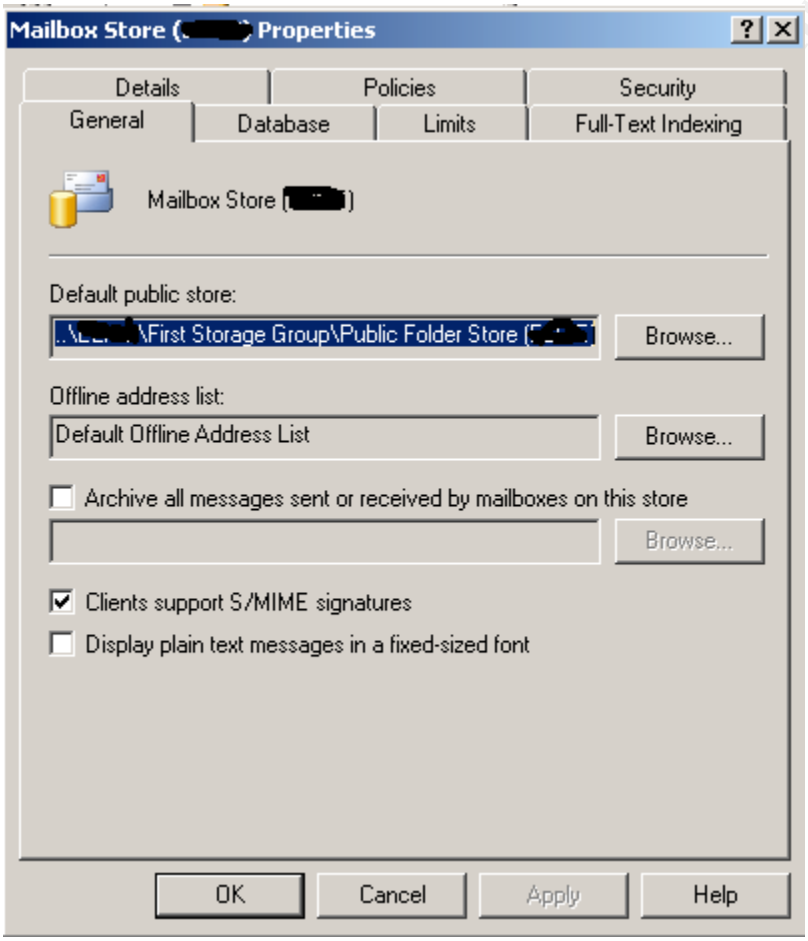
After clicking the Proxy Settings button this dialog appears. All the above are the recommended settings. If NTLM Authentication gives you problems, set that drop down box to "Basic Authentication" and since it's over HTTPS, your cleartext password will still be encrypted over the wire.



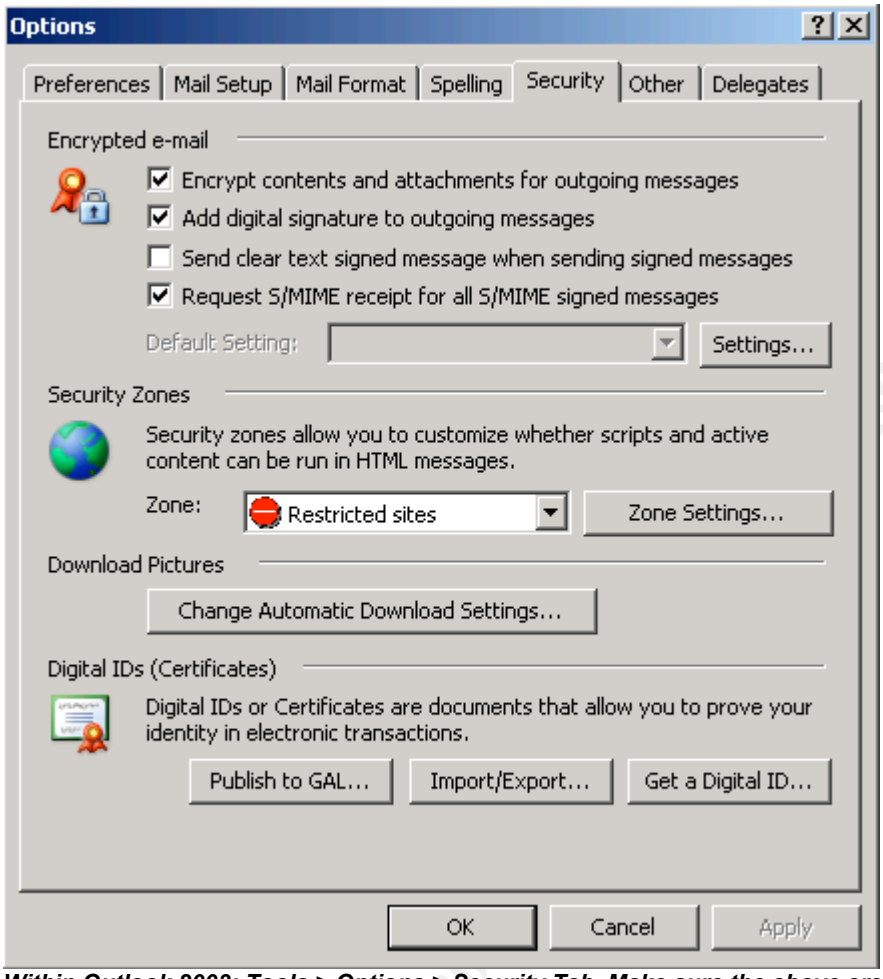
By running Outlook.exe with the /rpcdiag argument you can get this dialog which shows that I'm connecting over HTTPS.

Clients

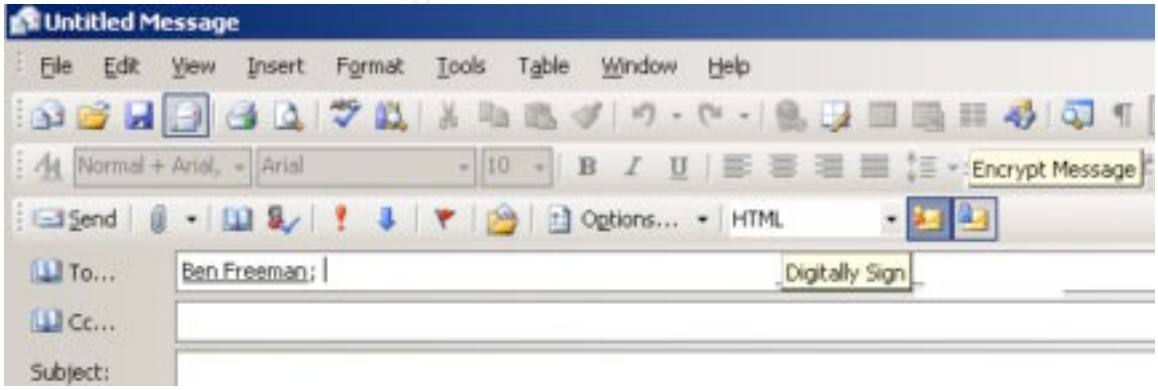
Inter-Office client security is already improved when implementing the above GPOs. All traffic between the Exchange Server and the MAPI clients is either Kerberos or RPC traffic and will be encrypted with 128-bit encryption. Since Kerberos is used as the authentication mechanism, your password is never sent over the wire either. This does not mean that emails sent outside your organization are encrypted. They are in fact not at all and can be read in plaintext by anyone in between your mail server and the destination mail server. S/MIME is one way of fixing that since it allows you to both digitally sign a message for identity proof and encrypt the message as well. This would mostly be used by people in high security positions where email contents are extremely confidential. Below I outline the steps required.



Make sure that box is checked on the Mail Store



Within Outlook 2003: Tools > Options > Security Tab. Make sure the above are checked.



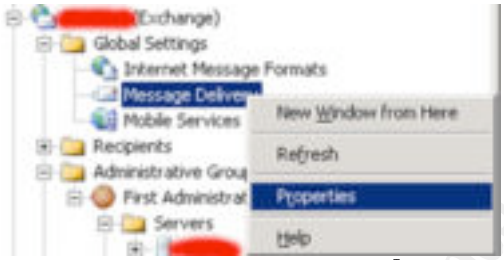
By default, once there is a user certificate within the Active Directory store, messages will be Digitally Signed and Encrypted. You can click either button to disable one or the other, or both.

Intelligent Message Filter

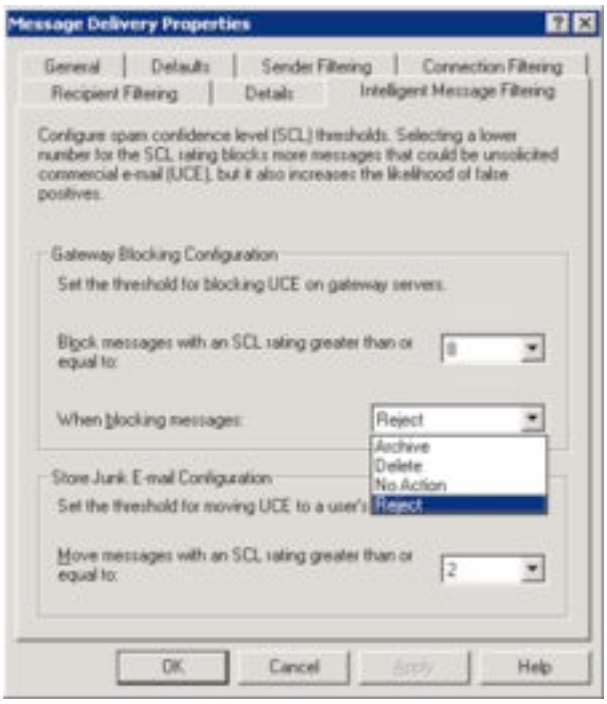
IMF is recommended to stop unwanted spam and viruses at the Exchange Server, before they get to users mailboxes. There are a number of different options for IMF to decide what spam is and what to do with the mail if in fact it

does meet the spam filter. Each message is given a SCL rating which is derived from the different fields within the message header and then given a rating of 1 to 10. As an administrator you can set the SCL rating of your IMF filter to anything between the ranges of 1 to 10 and then decide whether or not to filter it to the users Junk Mail folder or to just trash it. The reason I mention this plug-in is because most script kiddies first get their grips on the inside of an organization through email virus', Trojans, and backdoor scripts that innocent users open through their mail client. To eliminate this spam is a major improvement since you can prevent it from even getting to a users inbox. There are a lot of other products out there that can provide this functionality at a price, but I discuss this one because it is free. Below are some screenshots of the configuration and key benefits. The install is very straightforward so I won't show that part. Once installed, use the pictures below for configuration.

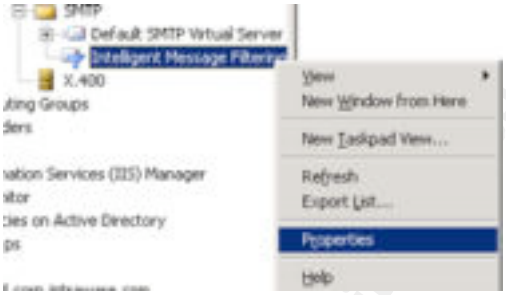
IMF Deployment Guide Download



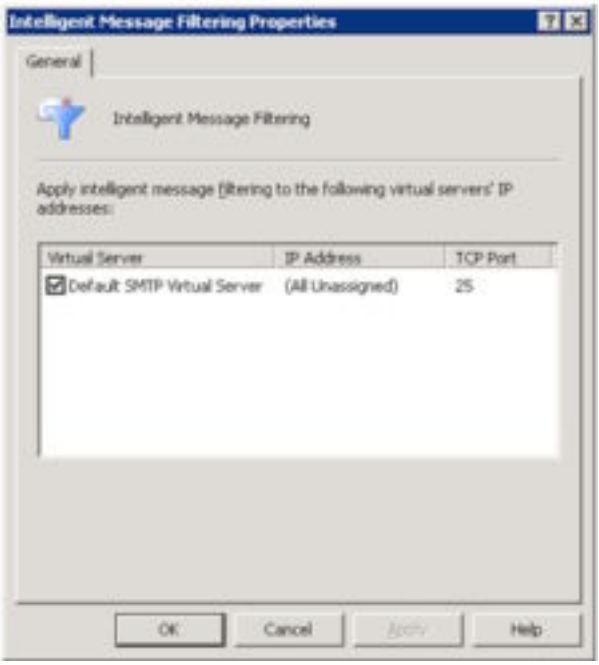
Within ESM go to Global Settings → Message Delivery → Right click, "Properties"



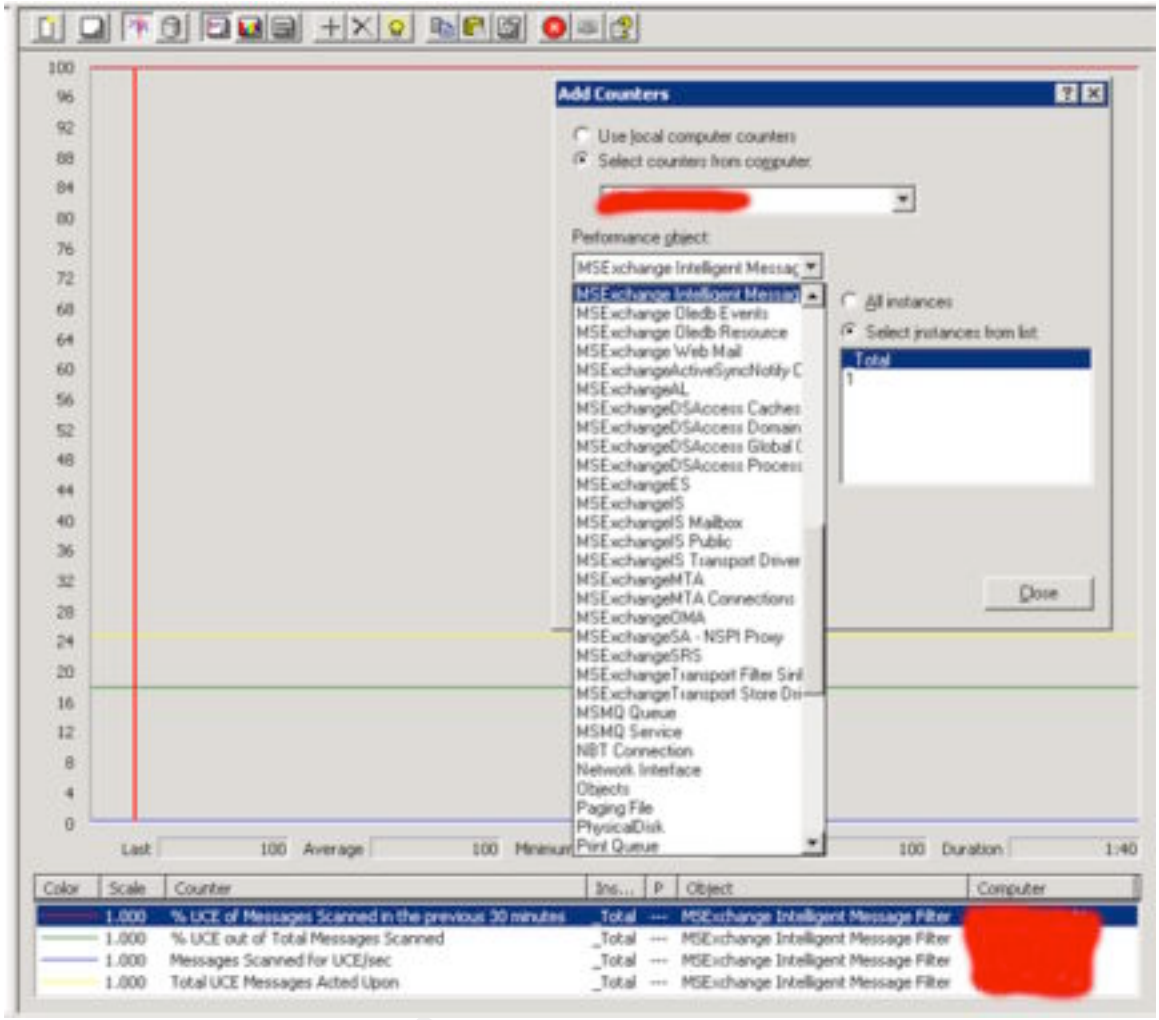
Here is the main IMF properties page. Above you can see that you can set the SCL rating for blocking messages from even entering the Exchange Org to begin with. You want to be careful about this setting since you don't want too many false negatives.



Within ESM, go to the protocol properties and then drill down within SMTP.



You'll want to make sure this is checked on all virtual servers you want IMF to monitor.



Once you install IMF, there will be counters added to the performance monitor on the Exchange Server. I want to point out the red counter at 100%. This is a huge deal, and shows just how much spam is filtered out with IMF. This can drastically decrease the chances of your users getting spam, and in return, viruses and Trojans.

Domain Controllers

The Exchange server will often talk to the domain controllers for LDAP info about user accounts and authenticating users for access to Exchange resources. You want to make sure all the GPO settings mentioned from above are set so that the RPC traffic between the two is encrypted. I also recommend installing certificates on the Domain Controllers for use with LDAP over SSL. That way LDAP information is never sent in cleartext. Your other alternative to LDAPS is to use IPSec for all traffic to domain controllers.

IPSec

IPSec is recommended for use if at all possible. Because this is a very complex topic, and the configuration is beyond the scope of this document it will

not be explained in detail. Also, there are no documents put out by Microsoft about the setup and configuration of this that I am aware of. There are a few considerations to include in the IPSec filters like terminal services (3389 default), SMTP (25), SSL (443), and LDAP (389). When configuring the IPSec filters keep in mind that the optimum situation is a FE/BE scenario. Because we are not in this situation we can not force IPSec for all communications. We are going to have to assume that every mail server in the world does not use IPSec and hence make it optional for at the very least port 25. Depending on your environment and the control over the types of clients connecting to the Exchange Server it may be possible to force IPSec communications on the remaining ports.

Other Considerations

Anti-Virus

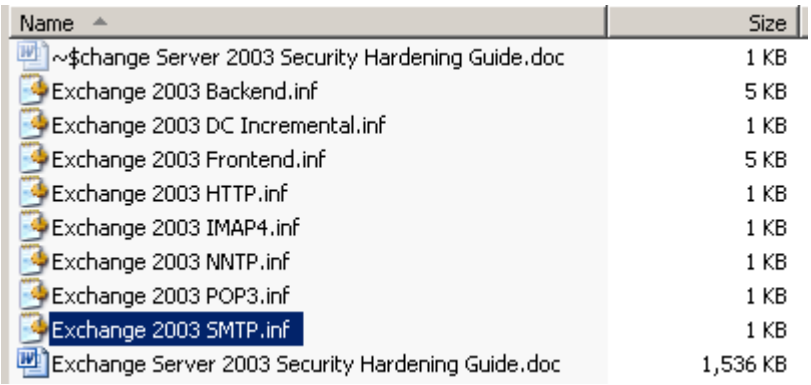
Anti-Virus directly ties in with IMF as far as reducing vulnerabilities through users. It is recommended to have a three-tier virus protection policy. The three scanners that should be in place are the firewall, SMTP gateway, and the clients. It also recommended having different vendors for client and SMTP gateway so that you have varying virus definition files in place. Sometimes one vendor will catch a new string before others will and this will give you the best possible protection from viruses. Another feature of the Outlook 2003 and the OWA clients are that they block many attachments by default so that users can not open them even if they want to. This is to protect from malicious files that traditionally get sent from hackers and script kiddies.

Firewall

There should be very few ports actually opened to the outside for an Exchange Server. Port 25 needs to be opened for SMTP to send or receive mail from external sources. For RPC over HTTPS and OWA you're going to need port 443 to be opened. These two ports alone will give you connectivity to receiving/sending emails outside of the organization, RPC over HTTPS, and OWA. Nothing else should be open to the outside on your Exchange Server.

Security Templates

There are a few recommended security templates from Microsoft and you can find tons more custom templates made by third parties and other Exchange Administrators. When you download the Exchange Server 2003 Security Hardening Guide there will be 8 Security Templates that come with it as shown below. These were taken from the "Exchange Server 2003 Security Hardening Guide[3]"



Name	Size
~$change Server 2003 Security Hardening Guide.doc	1 KB
Exchange 2003 Backend.inf	5 KB
Exchange 2003 DC Incremental.inf	1 KB
Exchange 2003 Frontend.inf	5 KB
Exchange 2003 HTTP.inf	1 KB
Exchange 2003 IMAP4.inf	1 KB
Exchange 2003 NNTP.inf	1 KB
Exchange 2003 POP3.inf	1 KB
Exchange 2003 SMTP.inf	1 KB
Exchange Server 2003 Security Hardening Guide.doc	1,536 KB

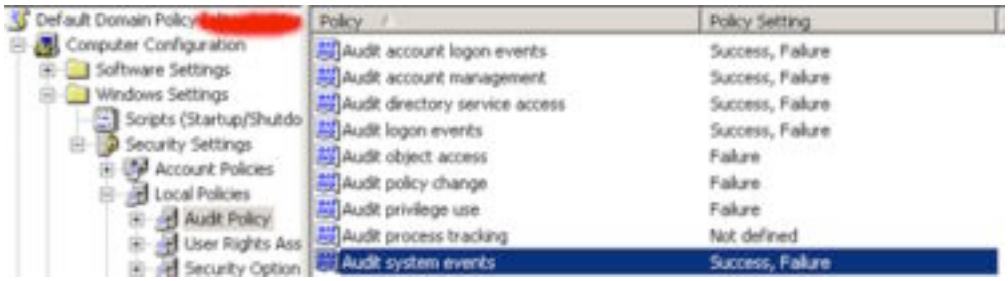
These templates are suggested to be used to open up the particular services as needed and are named appropriately. You'll want to look at the different templates in notepad or another text viewer before deploying it, and as always do so in a test environment first.
NSA Security Templates

Patch Management

Patch management is another extremely important aspect to making sure your Exchange environment isn't compromised. There are a number of different tools out there to reporting and pushing patches needed on a per machine basis. A couple free ones come straight from Microsoft and work pretty well. I **highly** recommend keeping up the latest patches since these are the attacks that will most likely be used first. Always deploy them in your test environment first to make sure nothing breaks and then schedule a time as soon as possible to get your production systems all patched up.
Microsoft Baseline Security Analyzer – Used for checking for common security misconfigurations
Hfnetchk – Checks for installed patches on any system

Auditing

You'll want to audit access to the server through a Group Policy Security Template for central management. This refers back to the templates mentioned earlier. You're going to want to audit all the events shown below to make sure you don't miss any critical forensic information. This information was taken from the "Windows 2003 Security Guide[1]"



Custom Scripts

Below I will display a script that I use to help me track failed event log messages. No admin can be watching event logs all day long and custom scripts should be an integral part of every administrator's toolkit. The script below will continuously monitor a systems local event log and email the interested parties the moment an error event occurs. This can easily be added as a startup script for all your critical servers through a GPO and prove very useful for keeping track of the important aspects of the event logs without checking them manually. Parts of this script were borrowed from "Windows 2000 Scripting Guide[5]"

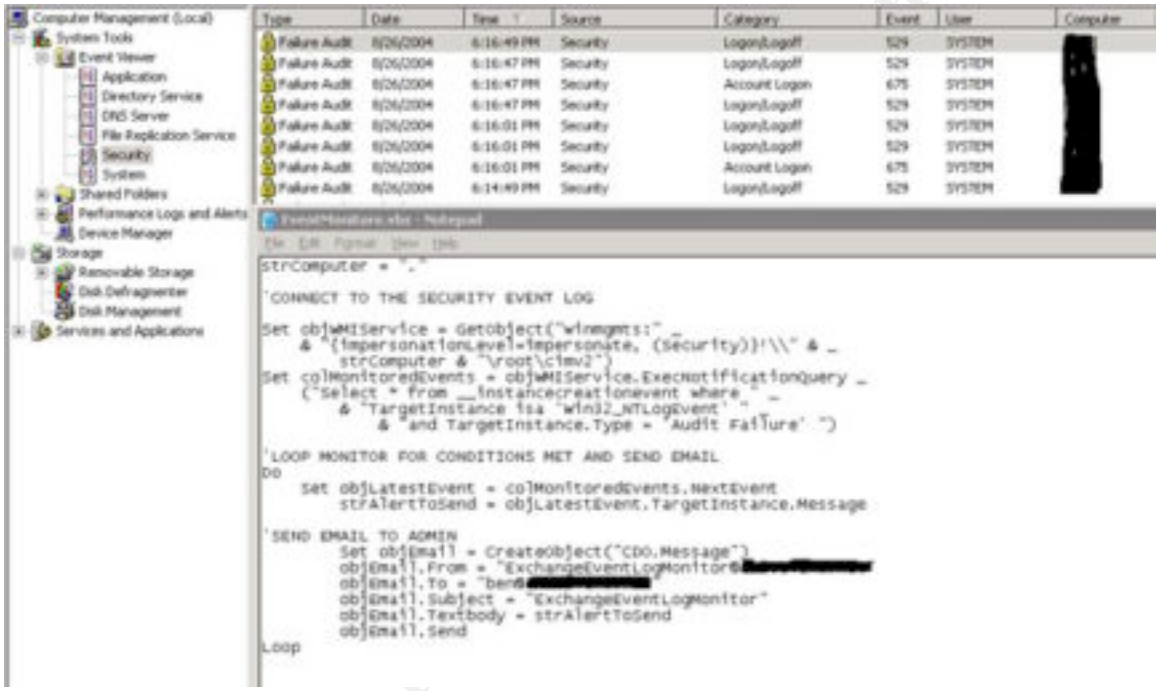
Script Code:

----------BEGIN SCRIPT----------

```
strComputer = "."

'CONNECT TO THE SECURITY EVENT LOG

Set objWMIService = GetObject("winmgmts:" _
   & "{impersonationLevel=impersonate, (Security)}!\\" & _
      strComputer & "\root\cimv2")
Set colMonitoredEvents = objWMIService.ExecNotificationQuery _
   ("Select * from __instancecreationevent where " _
      & "TargetInstance isa 'Win32_NTLogEvent' " _
         & "and TargetInstance.Type = 'Audit Failure' ")

'LOOP MONITOR FOR CONDITIONS MET AND SEND EMAIL
Do
   Set objLatestEvent = colMonitoredEvents.NextEvent
      strAlertToSend = objLatestEvent.TargetInstance.Message

'SEND EMAIL TO ADMIN
```

```
        Set objEmail = CreateObject("CDO.Message")
        objEmail.From = "ExchangeEventLogMonitor@domain.com"
        objEmail.To = "ADMIN@domain.com"
        objEmail.Subject = "ExchangeEventLogMonitor"
        objEmail.Textbody = strAlertToSend
        objEmail.Send
Loop
```

----------END SCRIPT----------



Script that monitors all Failure Audits in the security log and emails administrator immediately with the contents of the event log.

Conclusion

There is obviously a lot to think about as a small business trying to keep a single server instance of Exchange Server secure and still manageable. The steps outlined above should be a great step towards achieving this goal. A key factor for keeping your Exchange Server secure is going to be controlling access to it from remote locations. Windows Server 2003/Exchange Server 2003 has made big strides by locking down a lot of things out of the box compared to older versions. When utilizing the steps above for enabling SSL for OWA access and RPC over HTTPS you've taken large strides towards preventing information sent from remote locations from being compromised. Implementing assertive patch management, virus scanning, audit trails, firewall settings, spam filtering, remote access policies, password policies, and security templates is going to put you way ahead of the game against unauthorized attacks. Good domain security is

also imperative for the overall security of your Exchange Organization. There is no hacker kryptonite unfortunately, but at least when implementing the above recommendations you'll be one step closer to realistic prevention.

Appendix A: Links

Exchange Server 2003 Message Security Guide
http://www.microsoft.com/downloads/details.aspx?FamilyId=2305405C-FAF1-488A-A856-AD467BB59B26&displaylang=en

Exchange Server 2003 Security Hardening Guide
http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/exsecure.mspx

Windows Server 2003 Security Guide
http://www.microsoft.com/technet/security/prodtech/win2003/w2003hg/sgch01.mspx

Using Microsoft Exchange 2000 Front-end Servers
http://go.microsoft.com/fwlink/?linkid=4721

Exchange Server 2003 RPC over HTTP Deployment Scenarios
http://go.microsoft.com/fwlink/?LinkId=24823

Using ISA Server 2000 with Exchange Server 2003
http://go.microsoft.com/fwlink/?linkid=23232

Security Resources for Exchange Server 2003
http://go.microsoft.com/fwlink/?LinkId=21660

Exchange 2000 and 2003 - All Technical Articles and Books
http://go.microsoft.com/fwlink/?LinkId=10687

Exchange Server 2003 Client Access Guide
http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/cliaccgde.mspx

References

1. "Windows 2003 Security Guide", August 26th, 2004.
http://www.microsoft.com/downloads/details.aspx?FamilyID=8a2643c1-0685-4d89-b655-521ea6c7b4db&displaylang=en

2. Fossen, Jason – "Windows 2000/XP/2003 Active Directory" – SANS Institute, 2004

3. "Exchange Server 2003 Security Hardening Guide" August 2nd, 2004.
http://www.microsoft.com/technet/prodtechnol/exchange/2003/library/exsecure.mspx

4. "Exchange Server 2003 Message Security Guide", November 2004.
http://www.microsoft.com/downloads/details.aspx?FamilyId=2305405C-FAF1-488A-A856-AD467BB59B26&displaylang=en

5. Microsoft, Inc. – "Windows 2000 Scripting Guide" – Microsoft, Inc, 2003

6. "How to configure RPC over HTTP in Exchange Server 2003", Matt Kuhline, September 14th, 2004
http://hellomate.typepad.com/exchange/2004/01/how_to_configur.html

Last Updated: July 23rd, 2017

Upcoming Training

SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS vLive - SEC505: Securing Windows and PowerShell Automation	**SEC505 - 201709,**	**Sep 18, 2017 - Nov 13, 2017**	**vLive**
Secure DevOps Summit & Training	**Denver, CO**	**Oct 10, 2017 - Oct 17, 2017**	**Live Event**
SANS San Francisco Winter 2017	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**Live Event**
San Francisco Winter 2017 - SEC505: Securing Windows and PowerShell Automation	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**vLive**
SANS Cyber Defense Initiative 2017	**Washington, DC**	**Dec 12, 2017 - Dec 19, 2017**	**Live Event**
SANS Southern California- Anaheim 2018	**Anaheim, CA**	**Feb 12, 2018 - Feb 17, 2018**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**