

Realtime
publishers

The Shortcut Guide[™] To



Securing Your Exchange Server and Unified Communications Infrastructure Using SSL

sponsored by

GeoTrust® 

Don Jones

Chapter 3: Best Practices for Securing Your Exchange Server 32

 Business-Level Concerns for Exchange Security 32

 Securing Exchange Storage 33

 Securing Communications Channels..... 36

 User to Server..... 36

 Server to Server..... 44

 Server to Mobile Device 45

 Enabling Per-Message Privacy..... 45

 Best Practice: Layered, End-to-End Exchange Security..... 45

 Missing Pieces in the Security Picture..... 46

 The Social Channel: Educating Users..... 46

 Coming Up Next..... 48

 Download Additional Books from Realtime Nexus! 48

Copyright Statement

© 2011 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via email at info@realtimepublishers.com.

[**Editor's Note:** This book was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology books from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 3: Best Practices for Securing Your Exchange Server

In the previous chapter, we looked at general concerns and techniques around Exchange and messaging security. This chapter will focus specifically on Exchange, with a goal of securing every bit of the Exchange infrastructure. Not every organization will need to apply all the security we discuss here, but you should be able to pick out the pieces that matter to your organization and implement them appropriately.

Business-Level Concerns for Exchange Security

Before we start actually securing Exchange, we should briefly look at why we're doing so. After all, security is always a compromise between security and convenience. By applying any kind of security to Exchange, we're automatically making it at least slightly harder to manage, maintain, or use. Before doing that, we should examine our reasons for doing so to make sure the tradeoff is appropriate and necessary.

As I outlined in the prior chapter, the main business-level concern with Exchange is privacy, and there's often an additional need for integrity—making sure that key messages arrive without being tampered, and being able to authoritatively identify the messages' authors. Privacy is the big one, so that's what we'll be focusing on the most.

How we address that need for privacy depends primarily on where we feel our privacy threats originate. Most organizations worry the most about privacy being compromised *outside* the organization—that is, on the Internet. We'll definitely address that. Many organizations also worry about privacy being compromised *inside* the organization, which is a completely valid concern. Most intellectual property theft, for example, happens inside the company's office. Many laws and industry rules also focus on maintaining privacy within the organization.

You're going to need to know your organization's specific goals and requirements for security—ideally, communicated as a written security policy—before proceeding with this chapter. Knowing what your organization wants to achieve, from an Exchange security perspective, will let you know which sections of this chapter are applicable to you.

Securing Exchange Storage

Securing Exchange's message databases is the first thing we'll look at. Some companies don't worry about this, and if your organization exercises a high level of physical security for your servers, you may not need to worry. You worry about securing Exchange's physical storage when you're concerned about the risk of someone physically gaining access to the hard drives of an Exchange server. In larger environments, those hard drives may be part of a Storage Area Network (SAN), making it even more difficult for someone to physically take them (after all, with a SAN, you generally have to take *the entire SAN* in order to access anything on the physical drives).

However, some organizations aren't able to physically secure every Exchange computer, and some organizations are subject to external requirements (laws and regulations) that simply mandate a high level of physical security and redundant security. In those cases, you'll probably want to encrypt Exchange's database files.

Microsoft formally supports the use of Windows Server's BitLocker encryption technology to encrypt drives containing Exchange transaction log files and database files. You need to take care to encrypt *both* sets of files: Without encrypting the transaction logs, you're still leaving message data vulnerable to physical compromise. BitLocker doesn't impose a significant system overhead, and Microsoft spends a significant amount of time testing Exchange with BitLocker to make sure both perform well and are safe to use.

In a nutshell, BitLocker works by encrypting an entire disk. Typically, you'll encrypt the volume containing Windows as well as volumes containing Windows' page file(s) and Exchange's database files and transaction log files. On servers containing a Trusted Platform Module (TPM) chipset, BitLocker ensures that a computer's drives can only be used if the computer's critical startup components—BIOS, boot configuration data, and so forth—are unaltered. That makes it pretty difficult, if not impossible, to gain access to the protected drives' data. If they're not in their original computer, and if that computer isn't in a normal operating condition, the drives remain encrypted. As Figure 3.1 shows, that's really BitLocker's strength—because it relies on the TPM, any drives can't be accessed on their own. You'd have to steal the *entire server*—much less practical than ripping a few hot-swap drives out of an enclosure.

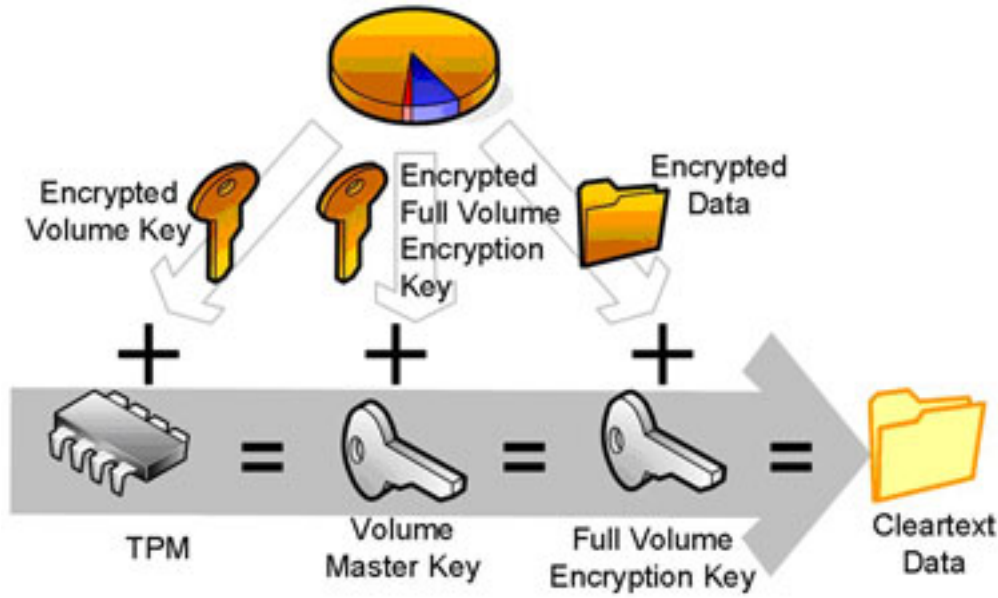


Figure 3.1: BitLocker functional diagram.

BitLocker also supports multi-factor authentication in computers that have a v1.2 or later TPM. This can, for example, lock the computer's boot process until someone enters a PIN or provides a specific USB flash drive. It's unusual to utilize multi-factor authentication on a server, though, because the need for an administrator to be physically present can compromise server recovery and other operations. BitLocker does require a separate, unencrypted partition of at least 1.5GB, which is used to store the Windows pre-execution environment (WinPE), boot files, and other startup files. This partition can be shared with tools and recovery utilities provided by the server's manufacturer.

BitLocker *is* compatible with hardware RAID solutions, which are commonly used in Exchange environments. However, BitLocker is only supported with drives and RAID arrays that are physically and exclusively connected to the Exchange computer; it *is not* compatible with SANs.

That said, many high-end SAN vendors offer their own encryption mechanisms that you could use in place of BitLocker and that operate below Windows' awareness. In some cases, for example, encryption is provided by using an encrypting SAN switch. Data sent to the SAN is encrypted by the switch, and remains encrypted on the SAN hard drives. Data retrieved from the SAN is decrypted by the switch and passed to servers. This can be used in conjunction with BitLocker: You'll still want to encrypt system volumes and volumes containing Windows swap files, for example, and those volumes will typically be local to the server, not stored on a SAN (an exception is virtual machines, where the operating system—OS—virtual drives might well be delivered from a SAN). Figure 3.2 illustrates some of the points where encryption could be applied.

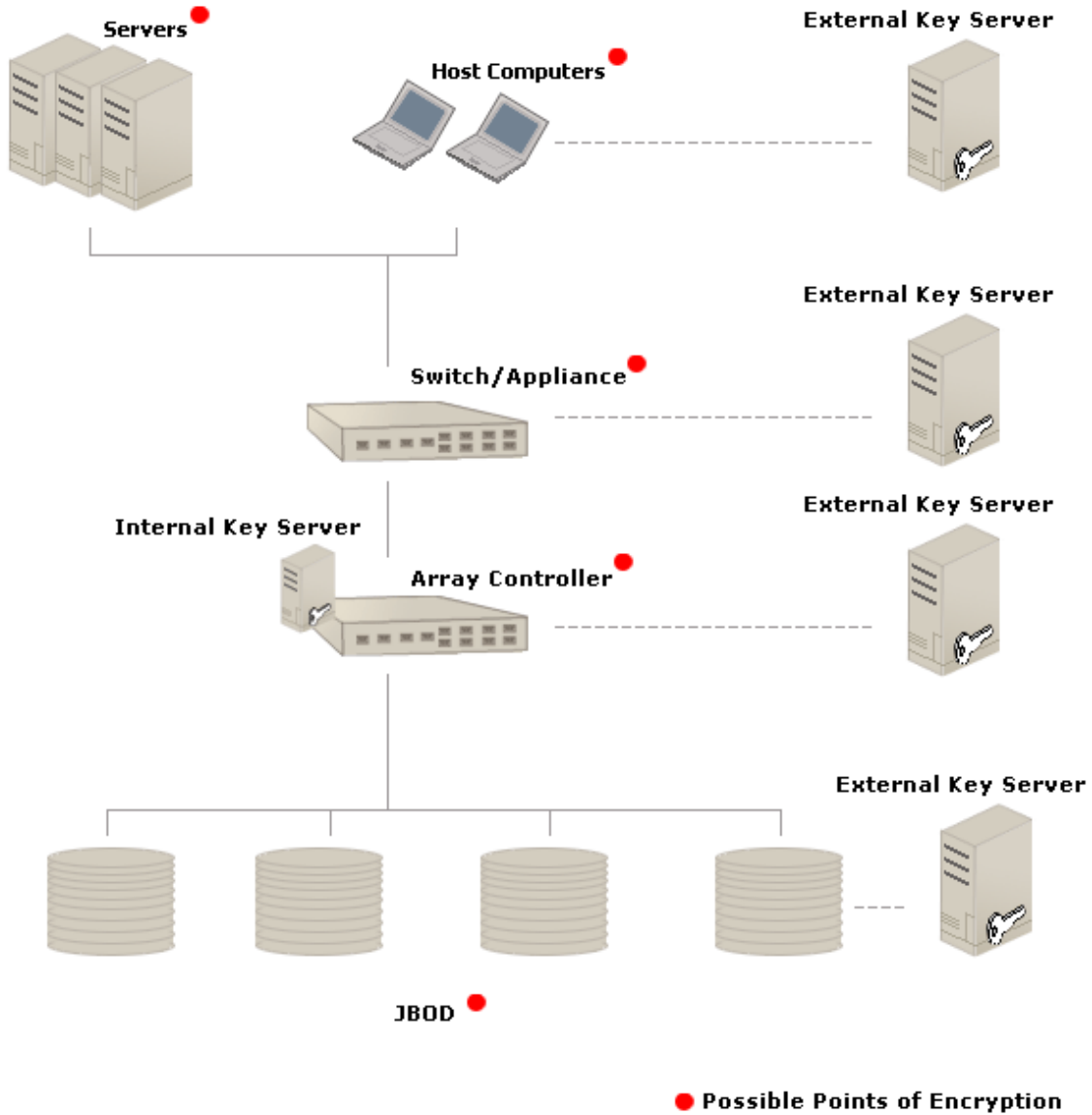


Figure 3.2: Potential points of encryption in a SAN.

As shown, it's also possible for server-based software to provide encryption for SAN data. This works pretty much like BitLocker, only designed for the SAN environment. Third-party software is required to make that happen. As illustrated, encryption keys—especially those used by an encrypting switch or array controller—can come from an external server, making the switch or controller useless if stolen.

Protecting a SAN in this fashion is effective because, like BitLocker, it stops someone from being able to access data by simply ripping a few hot-swap drives out of an enclosure. They'd have to take *every* drive in a RAID set, *and* figure out which SAN switch was providing the encryption and take that, too (as well as a key server, if one is used). In a decently-secured data center, that's pretty impractical. You obviously need to decide how far down this path *your* organization needs to go, but you can certainly provide all the physical security that Exchange might need using these techniques.

Securing Communications Channels

Once data leaves the Exchange server, it immediately becomes vulnerable again as it travels across the network. One option to help protect it is to enable per-message privacy, which I'll discuss shortly—but that only *helps*. Users access messages from a broad variety of devices these days, and simply encrypting individual emails doesn't protect every message in every scenario.

User to Server

Users can access Exchange messages through a variety of protocols. On their office network, users will tend to use Remote Procedure Calls (RPCs), which is the native protocol of Outlook. Users might also use HTTP and HTTPS, either in "RPC over HTTP" or "Outlook Anywhere," or by communicating with Exchange Web Services (the old Microsoft Entourage client software for Macs uses this, for example).

Externally, users might use IMAP, POP3, or HTTP to access their messages. In the case of HTTP, they could be using Outlook Anywhere with an Outlook client, or they could be using the Outlook Web App (OWA, also referred to as Outlook Web Access), the Web-based Outlook client.

Outlook 2007 and later automatically apply encryption to RPC connections, so we don't have to worry about that. That leaves us with three protocols to encrypt: HTTP, IMAP, and POP3. All support the use of SSL certificates for encryption. This is where Subject Alternative Name (SAN) certificates can come in handy. Rather than having to purchase and maintain an SSL certificate for each Exchange server—as well as whatever other Internet-facing servers you may have—you can simply purchase one SAN certificate to secure them all.

SAN: A Certificate Add-On

As described in Chapter 1, there isn't actually an entity known as a "SAN certificate." SAN is an option that can be added to most SSL certificates.

For example, high-end Certification Authorities (CAs) usually offer a "normal" SSL certificate as well as an Extended Verification, or EV, SSL certificate. The latter costs more and has a more rigorous identity verification, but also triggers more and better visual cues within Web browsers to help the user confirm the identity of the server they're connecting to. Both "normal" and EV certificates support the additional SAN field, enabling these certificates to identify multiple servers.

Again as discussed in Chapter 1, this *isn't* the same as a "wildcard" certificate. Those can protect an entire domain of computer names, but they don't feature the EV capability, although they do usually offer complete validation of the organization's information.

To begin requesting an SSL certificate, in the Exchange Management Console (EMC), right-click a Client Access Server (CAS), and select New Exchange Certificate (see Figure 3.3).

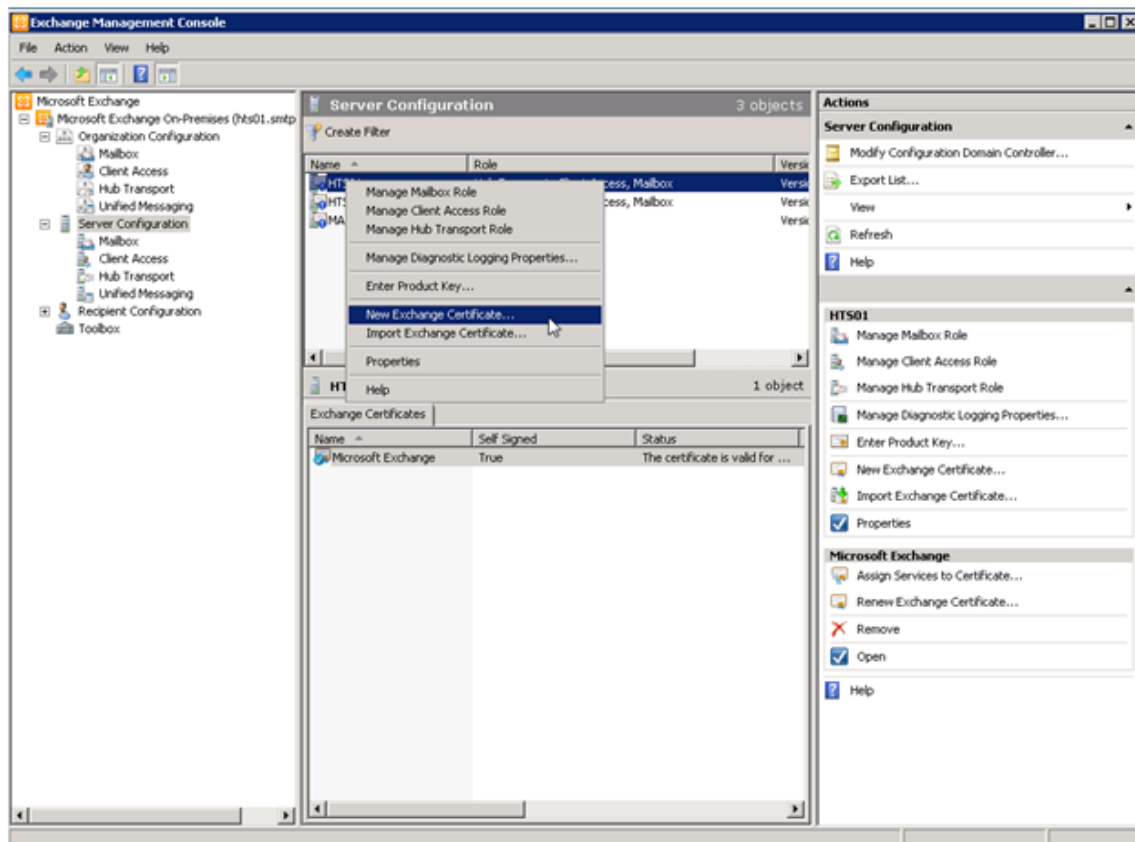


Figure 3.3: Starting a new Exchange certificate request.

You'll create a "friendly name" for the certificate, as Figure 3.4 shows. This can be anything you want—it's just a name to identify the certificate itself.



Figure 3.4: Certificate friendly name.

Don't enable the wildcard certificate. Remember, those don't support the SAN feature. Figure 3.5 shows the screen—just make sure to not select the check box.

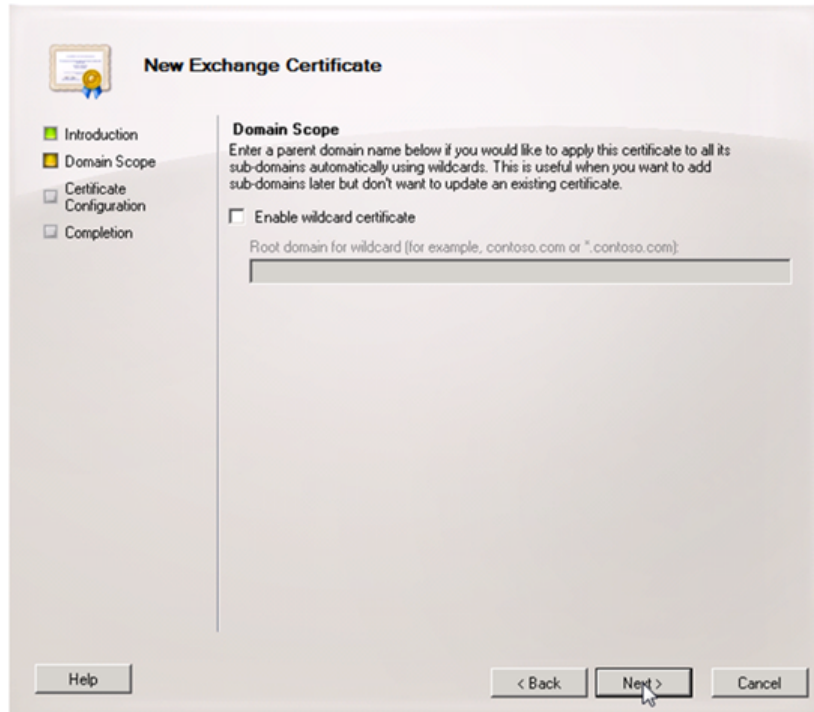


Figure 3.5: Don't enable wildcard certificate.

Next, as Figure 3.6 shows, fill out the certificate information. You'll need two host names here: the CAS server's *internal* name as well as its *external* name—the one outside users will visit to get to OWA.

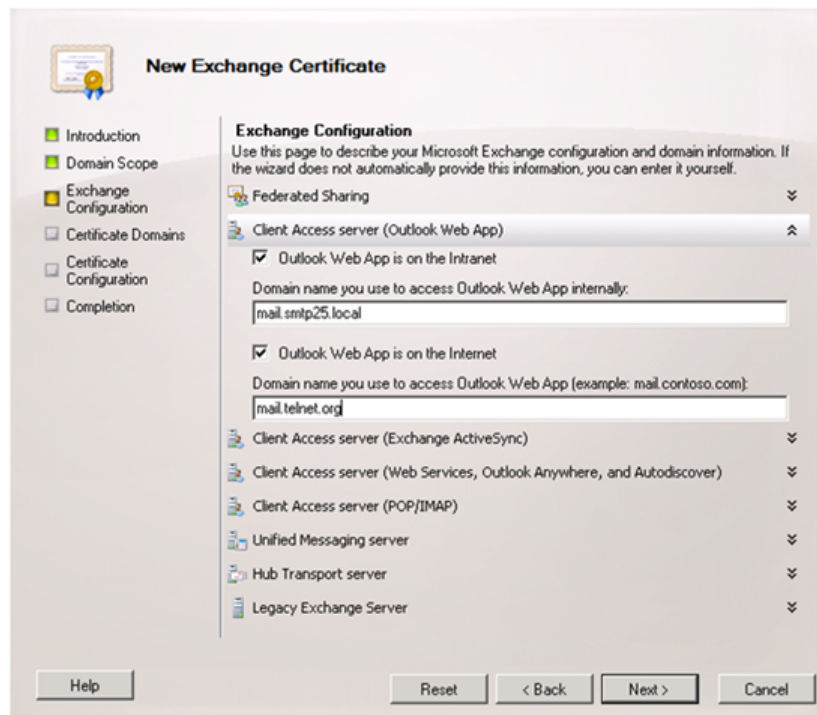
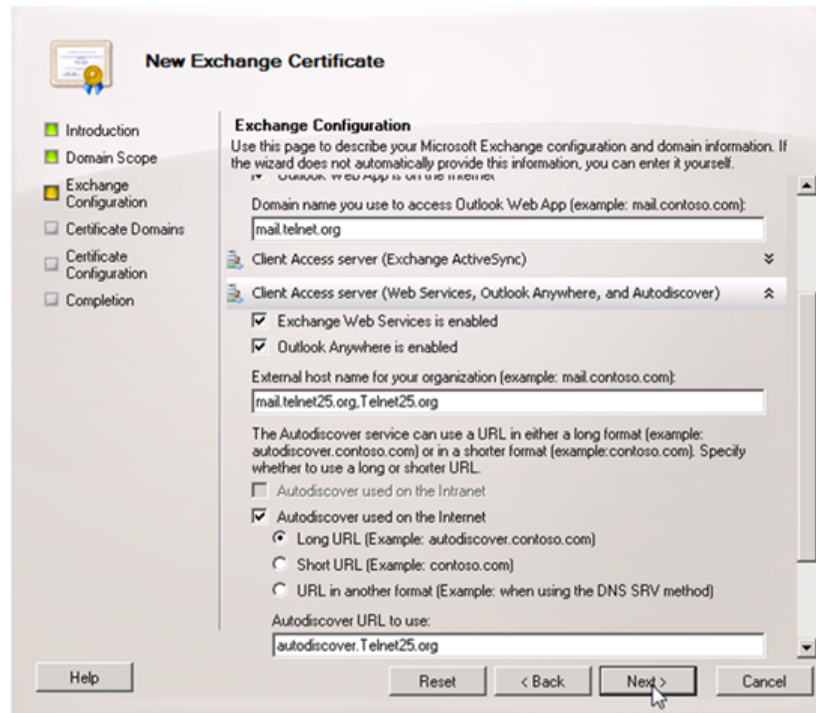


Figure 3.6: Specifying server names.

As Figure 3.7 shows, you'll then need to provide details about your CAS server, indicating which services are enabled so that they can properly use the certificate and listen on the right ports.



The screenshot shows the 'New Exchange Certificate' wizard in the Exchange Management Console. The 'Exchange Configuration' step is active, showing fields for domain name, client access servers, and service enablement options. The 'Domain name you use to access Outlook Web App' is 'mail.telnet.org'. The 'Client Access server (Exchange ActiveSync)' is selected. The 'Client Access server (Web Services, Outlook Anywhere, and Autodiscover)' is also selected. The 'Exchange Web Services is enabled' and 'Outlook Anywhere is enabled' checkboxes are checked. The 'External host name for your organization' is 'mail.telnet25.org.Telnet25.org'. The 'Autodiscover used on the Internet' checkbox is checked, with the 'Long URL' radio button selected. The 'Autodiscover URL to use' is 'autodiscover.Telnet25.org'. The 'Next >' button is highlighted with a mouse cursor.

Figure 3.7: Specifying Exchange services.

The next step, as Figure 3.8 shows, is critical: You need to specify the server's *common name*, which is what users will type into their Web browsers to get to OWA when they're *outside your network*. The screen will list the server's known host names; select the correct one, and click "Set as common name."

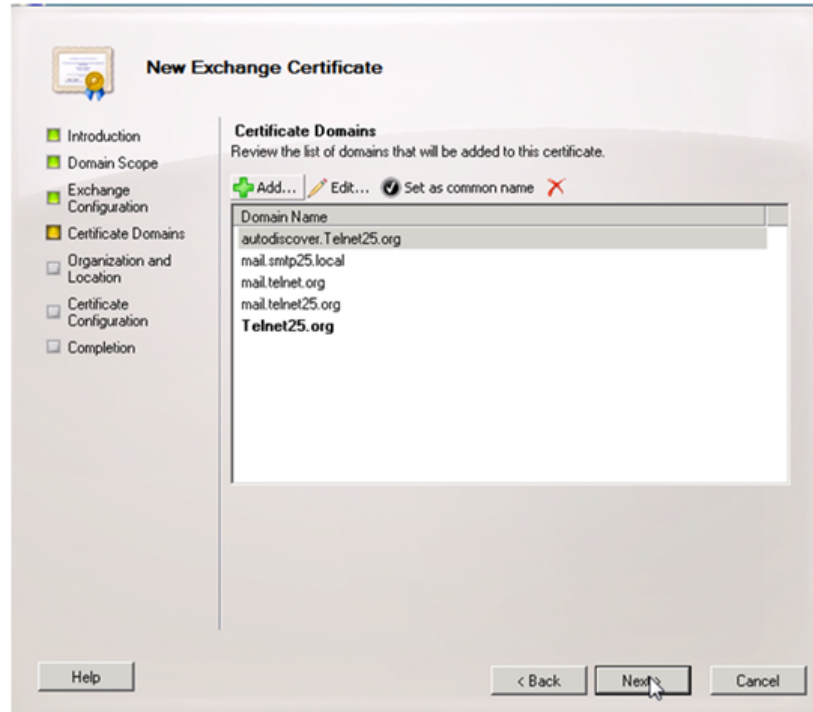


Figure 3.8: Setting the server's common name.

In Figure 3.9, you'll need to specify a filename to write the certificate request. Do so, and Exchange will create the request file.

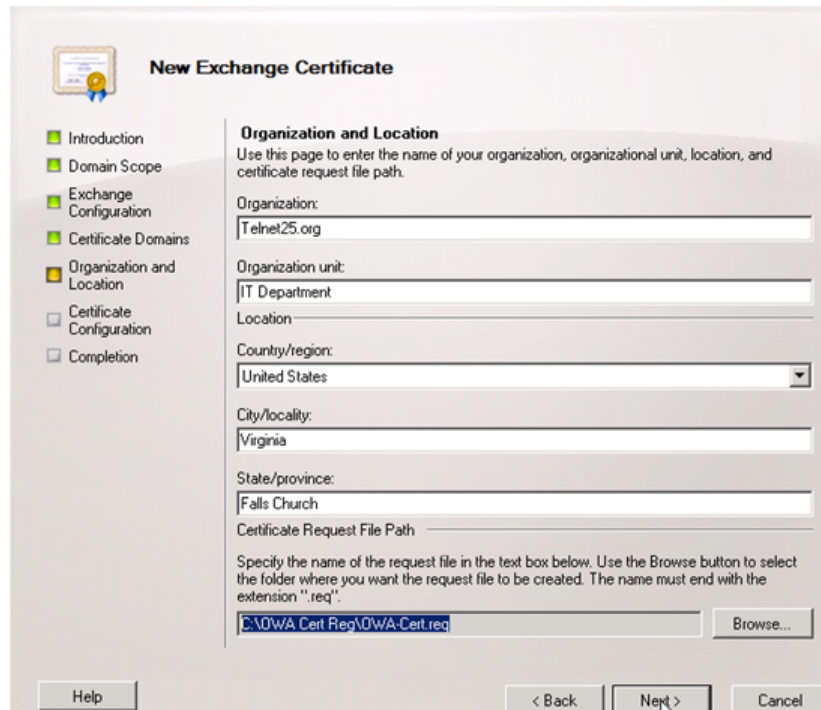


Figure 3.9: Saving the certificate request.

You can open the request file in Notepad, as Figure 3.10 shows, to review it—it’s just a bunch of letters and numbers. Some CAs will permit you to paste this information (which must include the START and END lines) into a Web site in order to obtain a certificate.

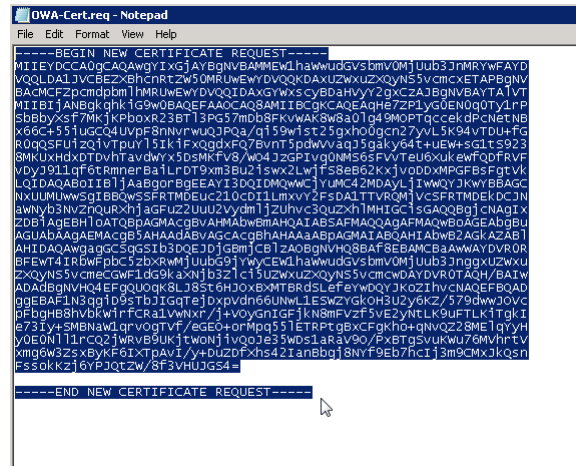


Figure 3.10: The completed certificate request.

Once the request is submitted to your CA, wait for it to be approved. With EV-class certificates, this will take additional time. I do recommend EV certificates because it’s easier to educate users to look for the EV-specific visual cues in browsers. If users aren’t educated properly (something we’ll discuss later), there’s no point in using a certificate at all.

Once your certificate is issued, you’ll download it, in a file, to the Exchange computer. Go back into the EMC, right-click the CAS, and complete the certificate request, as Figure 3.11 shows.

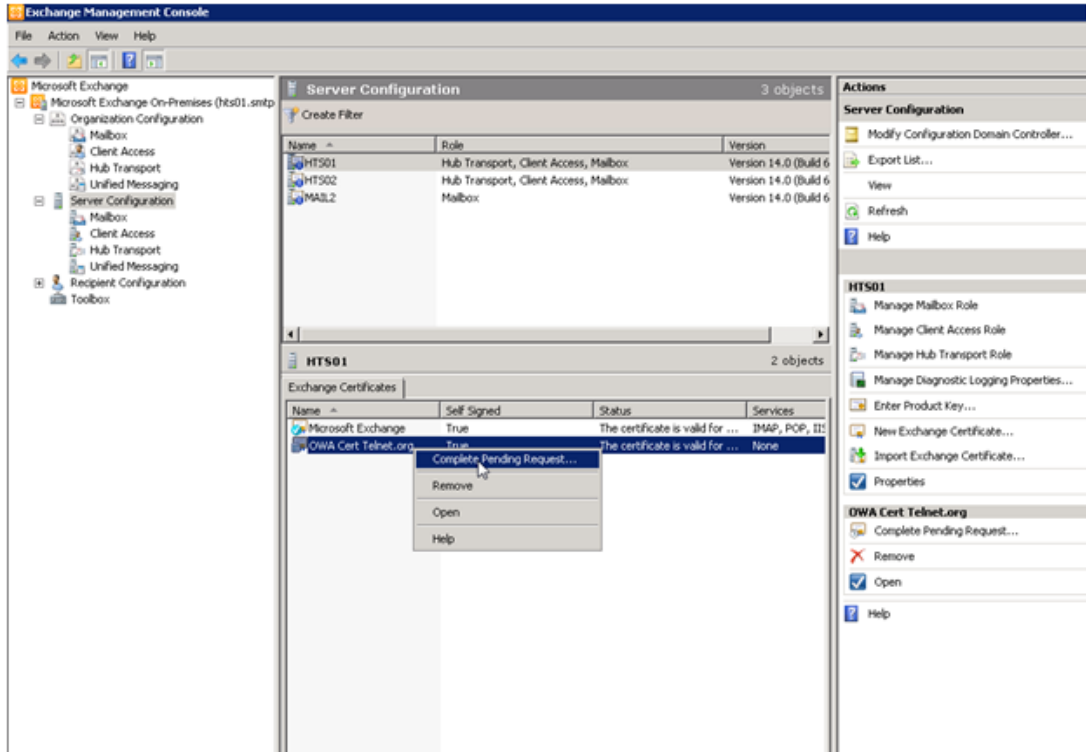


Figure 3.11: Completing the certificate request.

You'll simply browse to the downloaded certificate file, select it, and you're done. From there, you need to assign specific Exchange services to the certificate, as Figure 3.12 shows.

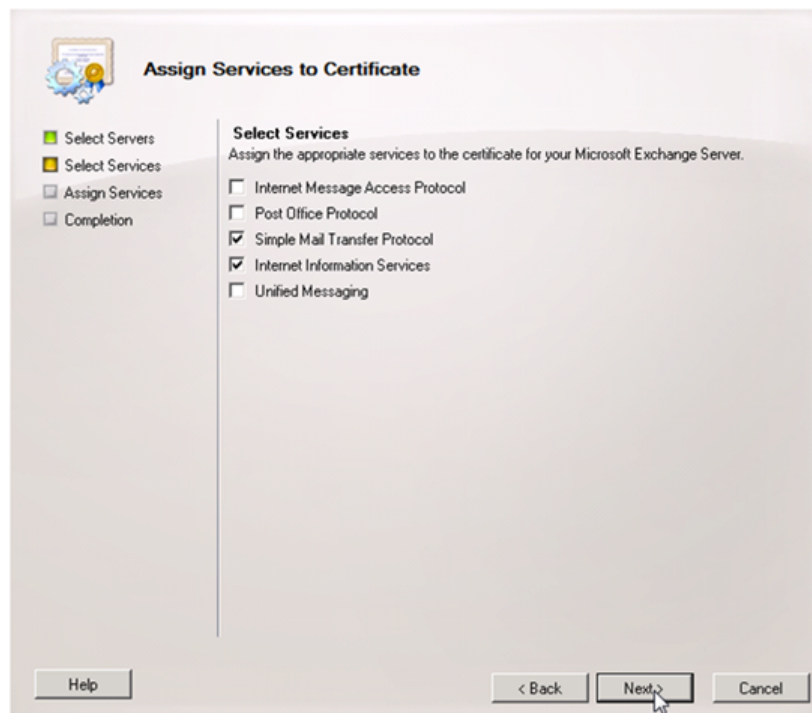


Figure 3.12: Assigning services to the certificate.

You'll notice that you can also protect the IMAP and POP3 protocols, in addition to OWA (IIS) and SMTP (which we'll discuss next). Finally, as Figure 3.13 shows, you'll have the option to *enforce* SSL, which prevents users from connecting with an unencrypted connection.

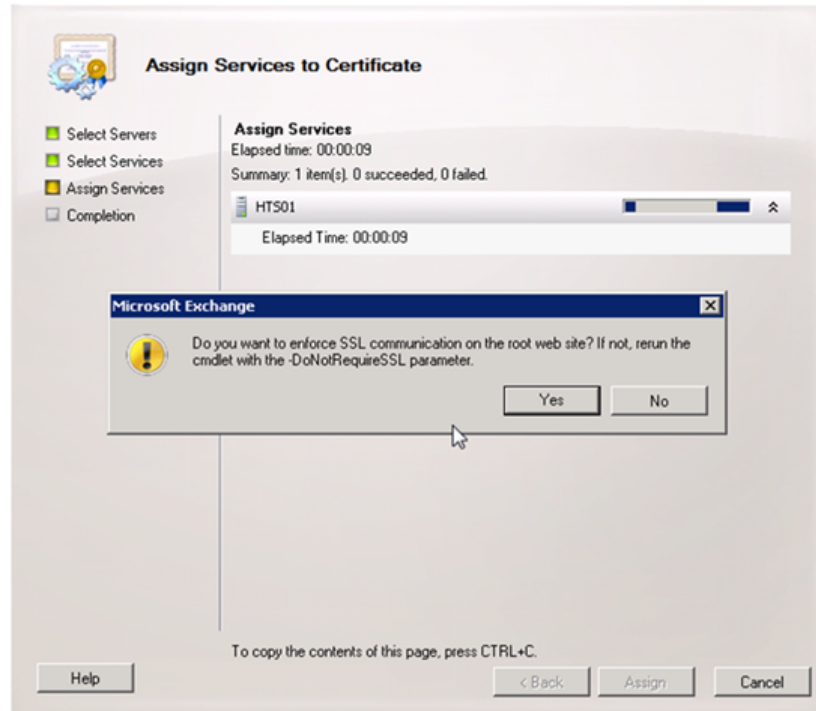


Figure 3.13: Enforcing SSL on the Web site.

Once that's done, you're finished: Communications via HTTP, POP3, and IMAP (depending on which ones you selected) will now be SSL-protected. You still need to take care to educate your users—we'll get to that in a bit.

Server to Server

Mail servers communicate with one another using SMTP, which can be SSL-protected as illustrated in the previous section. However, you have to exercise a bit of judgment here, and you have to recognize the shortcomings of the Internet in general.

You *can* enable SSL protection for SMTP on your own servers. You can even *require* that incoming connections be made only with encrypted SMTP; most modern mail servers are capable of negotiating that connection so that any mail you *receive* will be protected in-transit. You *could* also configure Exchange, in theory, to require SSL connections when it makes outgoing SMTP connections to send mail. In practice, however, you can't do that because you can't rely on every recipient's SMTP server being equipped with an SSL certificate. It's an unfortunate but true fact that the Internet's SMTP communications are largely *not* encrypted.

You also have no control over any kind of storage-level encryption employed by outside mail servers, so there's almost no point in asking them to provide an SSL-protected connection. If you can't secure the messages in-situ, there's almost no point in securing them in-transit, either. Your only option, if privacy of outgoing messages is a concern, is to provide users with per-message encryption, which we'll discuss shortly.

Server to Mobile Device

Mobile devices are a bit different. Any modern mobile device will support SSL-secured connections, so once you've set that up on your server, the communications channel will be secured. But mobile devices nearly always store copies of messages, and you're going to need to protect that storage. Because it's difficult, if not impossible, to remove the storage *from* the mobile device, you just need to protect the mobile device itself in the event it's stolen.

Exchange provides a means for doing so. When you set up an ActiveSync connection between a mobile device and Exchange, Exchange provides the device with a generated encryption certificate. That certificate can specify additional parameters, including the length of time a device may sit idle before locking itself, whether the device must require a PIN to be unlocked, and so forth. Setting these options provides a good level of security. Many devices even offer the option to wipe themselves if the wrong PIN is entered too many times, defeating brute-force attempts to guess the PIN, and most enterprise-class devices can be remotely wiped via ActiveSync if the user reports the device lost or stolen.

Enabling Per-Message Privacy

You can always choose to provide your users with email encryption certificates, enabling them to encrypt individual email messages. Keep in mind that, in order to encrypt a message, the *recipient* must also have a certificate because (as explained in the previous chapter) it's the recipient's public key that will be used to encrypt the message.

Best Practice: Layered, End-to-End Exchange Security

Exchange security works best when you practice "defense in depth," using a layered, multi-prong approach to security. Essentially, do *everything* I've described in this chapter. Secure Exchange's storage, secure communications between servers and clients, and enable per-message encryption by providing your users with encryption certificates.

In the end, you'll probably fall short on being able to protect *all* outgoing messages, simply because you can't practically require all recipient email servers to offer SSL for SMTP connections, and because you probably can't practically expect *all* recipients to have email encryption certificates. This is where organizational policy needs to step in and help users understand what information they can, and cannot, send to external recipients who can't receive encrypted email.

Doing Better with Security

I've been focused primarily on the native capabilities of Exchange in this discussion; if you're willing to bring in third-party software, you can sometimes enable better security and mitigate some of the shortcomings of the native capabilities.

For example, Pretty Good Privacy (PGP) is a set of products (now owned by Symantec) that are designed to provide enhanced email security. A gateway product provides encryption for all outbound email, for example, and can take the decision out of users' hands. Things like key management, particularly to external users, can still be a challenge, but third-party products exist to help address those challenges.

Missing Pieces in the Security Picture

We haven't discussed protecting client-level storage, and plenty of email ends up with copies on client computers. Particularly with laptops, which are much more likely to be lost or stolen, you need to decide how you want to deal with this risk. One option, of course, is BitLocker, provided your version of Windows supports it. BitLocker should be configured to use a PIN or other multi-factor authentication method, because without it BitLocker is pretty pointless (someone who steals the whole laptop can simply boot it, and BitLocker will let them in unless it's configured to ask for a PIN, require a USB key, or need some other factor). Users *can* password-protect their offline storage files in Outlook, but Outlook opens those automatically each time the application starts, making that a minimally-effective form of protection against theft of the entire computer.

The Social Channel: Educating Users

Getting your users to *use* your security features is, of course, the challenge. Things like storage-level encryption can be made pretty much invisible. If you're going to rely on per-message encryption, users will need to be educated in how and when to use it. For example, they'll need to know the best way to obtain a recipient's public key (so that encrypted messages can be sent) is to ask that person to send a digitally-signed message because that will contain their public key.

A big area of education is in certificates used to protect OWA. HTTPS only works when your users *know to make sure it's working*. In other words, if your users aren't checking, their browsers—especially once outside your corporate network—can be easily tricked into visiting an imposter site.

Train users to *not* accept certificates just because their browser displays a pop-up error message like the one shown in Figure 3.14. Their browser is trying to tell them that something is *wrong*, and clicking "Yes" is probably not the right answer.



Figure 3.14: Browser certificate warning.

What's more, train them out of the "no news is good news" mentality. That is, just because they're not seeing a warning message from the browser isn't sufficient. They need to explicitly look for visual cues that tell them they're connecting to the right server. Consider Figure 3.15, which shows three Web sites being accessed in a recent version of popular browsers.

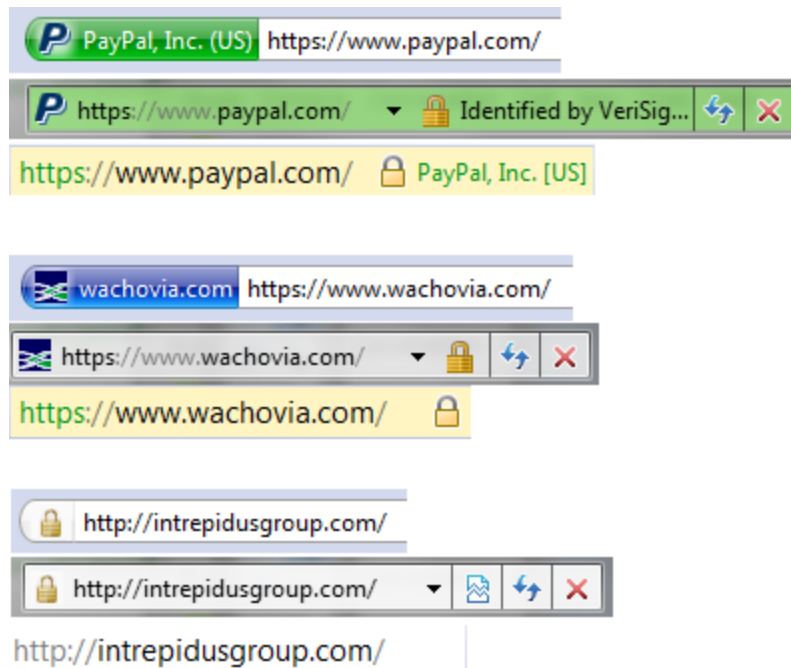


Figure 3.15: Composite view of browser SSL cues.

The bottom set is a Web site that isn't protected by SSL—notice that the URL starts with “http” and not “https.” No protection, and no guarantee that the browser isn't being fooled. In that set, you're looking at three browsers' address bars.

In the middle is a normal, secured HTTPS connection—the URL starts with “https” and two of the browsers show a “lock” after the URL. This indicates a normal SSL certificate, and the server's identity is confirmed and the connection encrypted.

My preferred approach is illustrated in the top set, which shows an EV certificate. Notice that the visual cues are much more significant: Bright green, the company's name (as opposed to just the domain name), and other cues make it much clearer that this server's identity is validated. This is the benefit of an EV certificate (which also supports the SAN option to protect multiple servers with a single certificate).

Coming Up Next

The final chapter in this guide will cover best practices for securing a unified communications (UC) infrastructure built around Microsoft's UC products, including Lync. As in this chapter, we'll look first at business goals, then move through the various layers of the infrastructure with best practices.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.