Realtime publishers

The Shortcut Guide™ To

Securing Your Exchange Server and Unified Communications Infrastructure Using SSL

sponsored by

GeoTrust®

Don Jones

Copyright Statement

Chapter 4: Best Practices for Securing Your Unified Communications Infrastructure

Exchange Server isn't the only thing we're using to communicate these days. Increasingly, users are also relying on highly-integrated instant-messaging tools, such as Microsoft Lync. Originally valued as a way to quickly send short, informal messages to colleagues in real-time, these tools quickly became a synchronous form of email (including email-like features such as file attachments). Voice and video capabilities came next, and instant messaging clients began to replace phone calls. At the same time, unified communications and unified messaging efforts connected traditional phone systems to Exchange, making a user's "unified inbox" a repository for email, voicemail, and other forms of communications.

As our options for communicating have expanded, so have the ways in which information can be improperly disclosed. It's time to make sure we have a firm grip on what our security goals are for unified communications (UC) and to make sure we're implementing measures to achieve those goals.

Business-Level Concerns for Communications Infrastructure Security

Our security goals typically look a lot like they did for Exchange itself. As with Exchange security, broader communications security is a compromise: 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. The main reason, of course, is privacy, just as it was for Exchange.

UC can enable users to feel a false sense of security. For example, despite what Hollywood would have you believe, wiretapping is actually quite rare in most countries, and your phone conversations are likely to remain private. Leave a voicemail, however, and it isn't necessarily in the hard-to-reach depths of the phone company: These days, it's just as likely to be a voice recording stored in Exchange Server as a message attachment. Move your voice conversation to Microsoft Lync, and you're off the telephone wires completely and instead pumping Voice over IP (VoIP) over your wired and wireless networks. Those communications are actually *much* easier to tap into. Retrieving your voicemails from email using a mobile device in a coffee shop? Easy to intercept. That's why security is so important across the entire communications infrastructure: Users may instinctively be less cautious because they're using systems and interfaces that *seem* inherently more secure.

Best Practices for Securing Lync Server

Lync Server can be especially complex to configure because it uses so many Web-based endpoints: there's one for Web services, one for the DialIn Simple URL, one for the Meet Simple URL, and so on. Although all of these can use a single IP address, they all need unique host names—and that means they each need a unique SSL certificate, or need to be named in a Subject Alternative Name (SAN) certificate. In fact, Microsoft recommends the use of SAN certificates, with the certificate's common name pointing to Lync Server's Web services URL, and SANs for the Meet Simple and DialIn Simple URLs.

Configuring IIS to use these certificates (or single SAN certificate) is straightforward, and is essentially similar to the way you configured OWA to use SSL in the previous chapter. It's actually fairly uncommon (and not recommended) to put Lync Server directly on the Internet. Instead, you'll nearly always have it behind a firewall—which results in more complexity. With Microsoft's ForeFront Threat Management Gateway (TGM), for example, you'll need to create publishing rules that explicitly permit SSL back to Lync Server (you'll often do something similar for OWA on Exchange). So let's walk through that configuration.

First, you'll need to get the SSL certificates properly installed in IIS on the Lync Server itself. Then, in TMG (if you're using a different firewall product, then this process will be logically similar), right-click the firewall publishing rule node, and select New | Web Site Publishing Rule (see Figure 4.1).

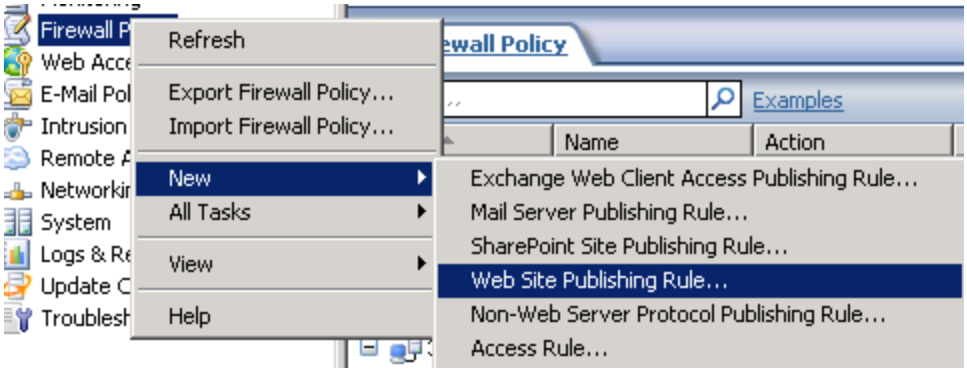

Figure 4.1: Creating the new publishing rule.

You'll need to configure TMG to forward the originally-requested host header. Doing so enables IIS to see which host the original HTTPS request was for so that the appropriate Lync Server Web site can be handed the request. Lync Server should believe that the requests are originating from the firewall. Figure 4.2 shows this configuration.

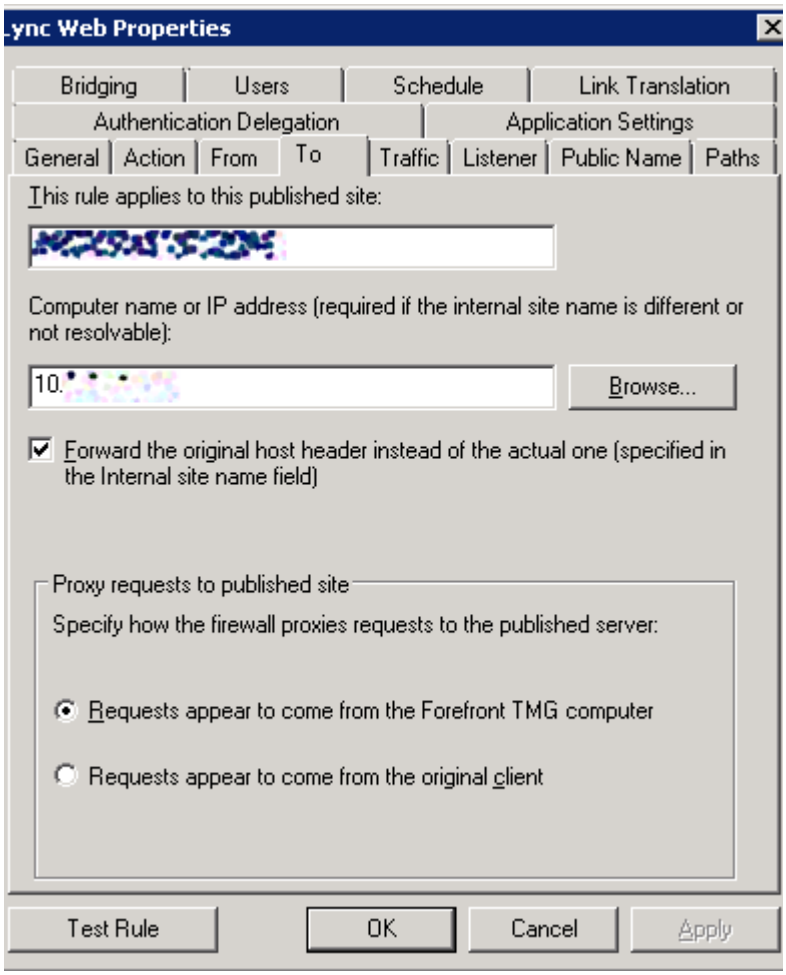

Figure 4.2: Configuring the publishing rule destination.

Next, tell TMG what to listen for. It's critical that both HTTP and HTTPS connections be permitted, as Lync Server expects both. However, you usually do *not* engage HTTP-to-HTTPS redirection: Lync Server will generally manage that. However, if you want to ensure that only HTTPS communications are used, you can enable the redirection of all HTTP traffic to HTTPS. Figure 4.3 shows the configuration.

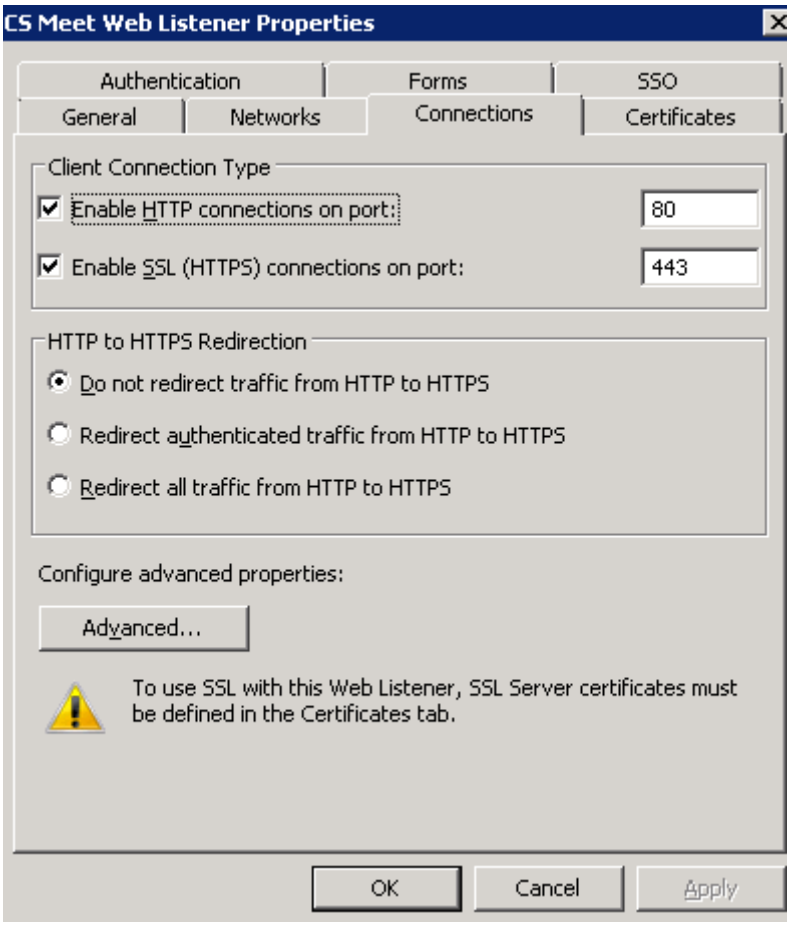

Figure 4.3: Configuring incoming connections.

On the Bridging tab, configure TMG to bridge incoming requests to the appropriate ports, as Figure 4.4 shows.

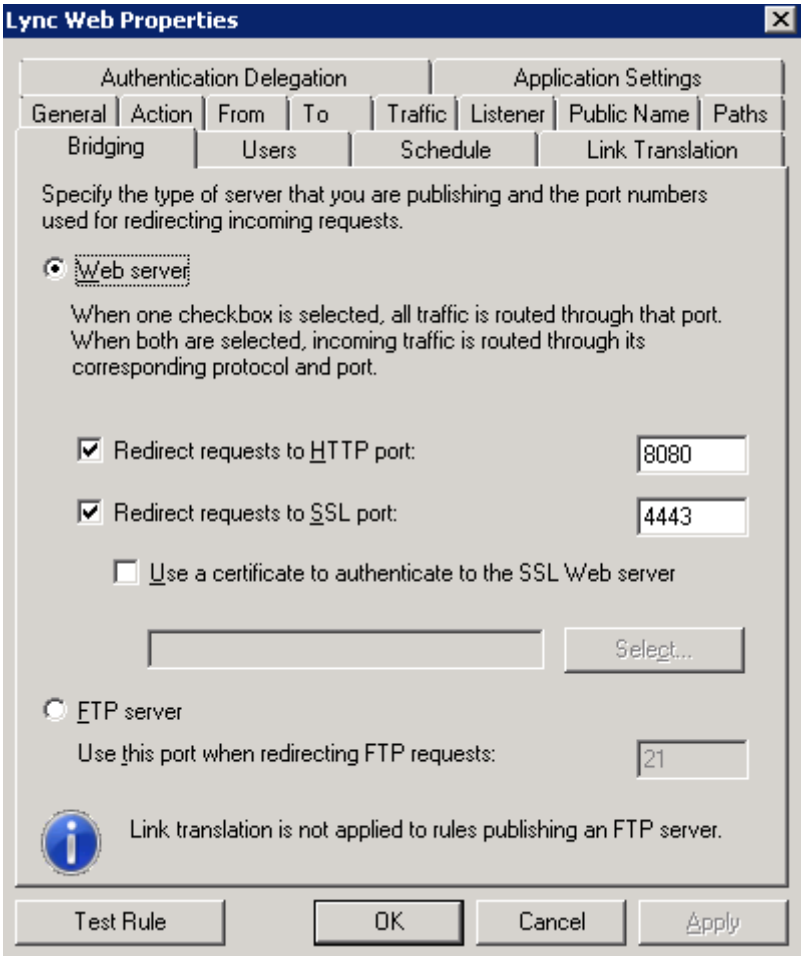

Figure 4.4: Configuring incoming request bridging.

Finally, you'll need to configure the public names that TMG will be listening for. These are the host names that appear in your SAN certificate and are configured in IIS on the Lync Server itself. Figure 4.5 illustrates a configuration, with three hosts named "dialin," "lyncweb," and "meet."

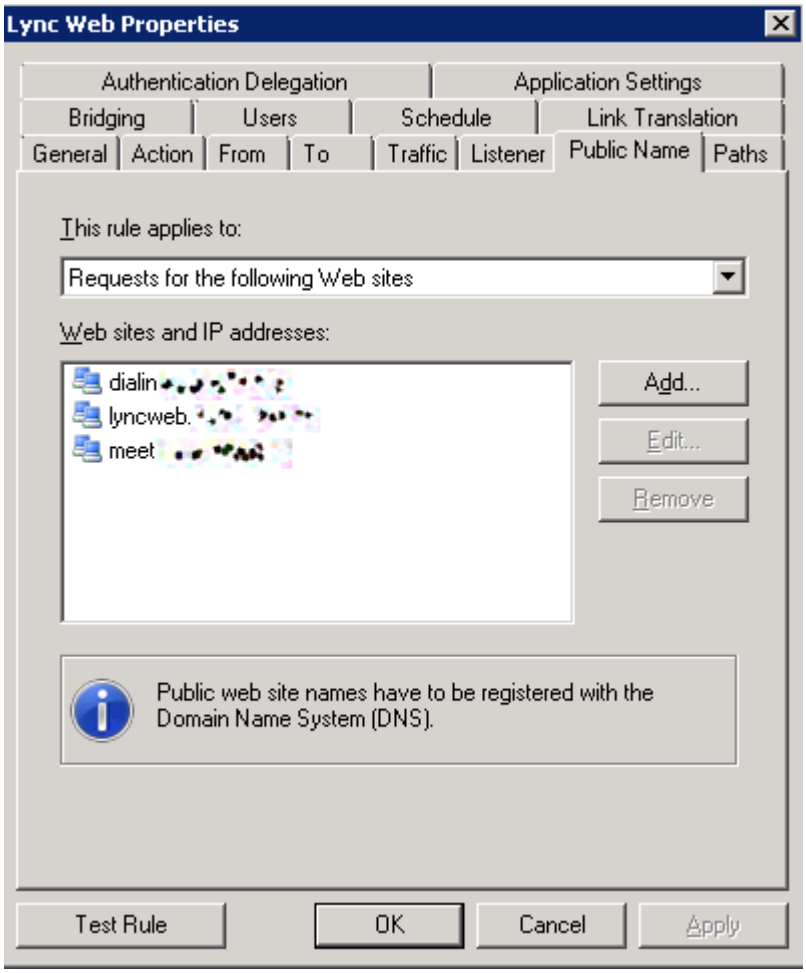

Figure 4.5: Configuring the public host names.

Once that's done, you should be good to go. Lync Server will enjoy the added protection of the firewall (including protection against Distributed Denial of Service—DDoS—and other attacks), and your communications will enjoy the privacy afforded by your SSL certificate.

There's one other area where a certificate will come into play, and that's when you're configuring the link between Lync and Exchange for Unified Messaging (UM). In those cases, be sure to configure Exchange *not* with a self-signed certificate but rather with a *trusted* certificate issued either by an internal or external Certification Authority (CA).

Resource

You'll find a deeper description of this CA configuration process at http://blog.schertz.name/2010/11/lync-and-exchange-um-integration/.

Lync <> IM

I'm often asked how to secure communications between Lync-connected instant messaging clients, such as Microsoft Office Communicator. The short answer is that "you don't." Those IM communications are largely peer-to-peer; Lync Server merely provides the *presence* information so that the IM clients can find each other. Every message sent doesn't flow through Lync, so there's no specific configuration on Lync that you'd use to secure those communications. That said, the IM clients do have their own configuration and can (in most cases) be configured to encrypt communications. Because of the variety of clients out there, that discussion is beyond the scope of this guide. Briefly, both clients intended for use with Lync Server—that is, Microsoft Lync for Windows and Microsoft Office Communicator for Mac—support encryption using TLS/SSL. The mobile versions—Microsoft Communicator Mobile, available for various mobile phone platforms—also provide encryption support.

Best Practices for Securing Mobile Devices

Mobile devices remain amongst the trickiest of the client components in the communications infrastructure. They're extremely mobile, prone to theft and loss, and typically don't support all of the powerful, modern security features we associate with our bigger laptop and desktop computers. Microsoft's effort at making mobile device management easier, more consistent, and more secure is System Center Mobile Device Manager (SCMDM). It's still an early-stage product, supporting a limited number of devices at present, although more are coming (most Windows Mobile or Windows Phone 7 phones released after 2009 are supported to at least some degree, along with numerous non-Windows devices). SCMDM does a lot; this discussion will focus on its integration with the communications infrastructure—specifically, Exchange Server.

SCMDM's main feature set enables it to:

- Prohibit the use of selected communications protocols on devices, such as disabling email entirely.
- Remote-wipe lost or stolen devices, destroying any locally-stored data.
- Connect mobile devices to the corporate network via an SSL-encrypted Virtual Private Network (VPN). This is a great "catch-all" way for securing traffic that can't otherwise be secured. Mobile devices that permit tethering (that is, using the device as a sort of cellular modem) can extend this VPN to tethered laptops or other computers, further enhancing the available security.

The remote wipe is perhaps the must-have feature. In fact, smart organizations today are denying users the option to use any device that can't be configured with remote-wipe functionality of some kind; any "enterprise-ready" device includes it (that includes Apple's iPhone). Remote wipe deals with one of the primary problems of mobile devices, which is that they typically support secure *connections* but not secured *storage,* so anything stored on them is at-risk if the device is stolen or lost.

Best Practices for Other Communications Channels

What other communications does your organization need to secure? We've looked at email extensively in the previous chapter, and this chapter touches on some of the security aspects of Lync Server and instant messaging in general. The real question to ask is *how else do your users communicate?*

Any form of communication can be intercepted, tapped, recorded, and examined; encryption is typically the best defense because you can't stop any of those activities. When communications are flowing digitally across a public network, you'll typically find SSL certificates involved to provide that encryption. When an organization has multiple communications endpoints, SAN certificates can provide an easier-to-manage means of getting that SSL encryption in place on all of those endpoints.

For example, perhaps you don't want your users sending file attachments through email simply because you *can't* always ensure the security of Internet SMTP connections and server storage, and you can't easily force users to encrypt individual messages (and there are reasons, discussed in the previous chapter, why you might not want to). In that case, you might provide your users with a Managed File Transfer (MFT) system to handle ad-hoc, person-to-person file transfers. That MFT system becomes another communications channel, and should be secured. In the case of MFT, you'll usually find options for transmitting files via some flavor of secured FTP, most of which utilize—you guessed it—SSL.

The key is to look at *all* of the ways in which information flows to, from, and within your organization and apply a logical, holistic security policy. Which, in fact, brings us to our next point.

Best Practice: Layered, Holistic Communications Security

That next point is actually the main point I want to make in this chapter: Take a look at your *whole* communications infrastructure, and look at the areas that need to be secured. You can do so in phases. For example, start with the obvious: email. In the diagram that Figure 4.6 shows, red lines indicate potentially-risky communications links. Red circles are storage points potentially under your control, and the red "X" is a storage point outside your control. The broken red line is a communications link that you can't necessarily control.

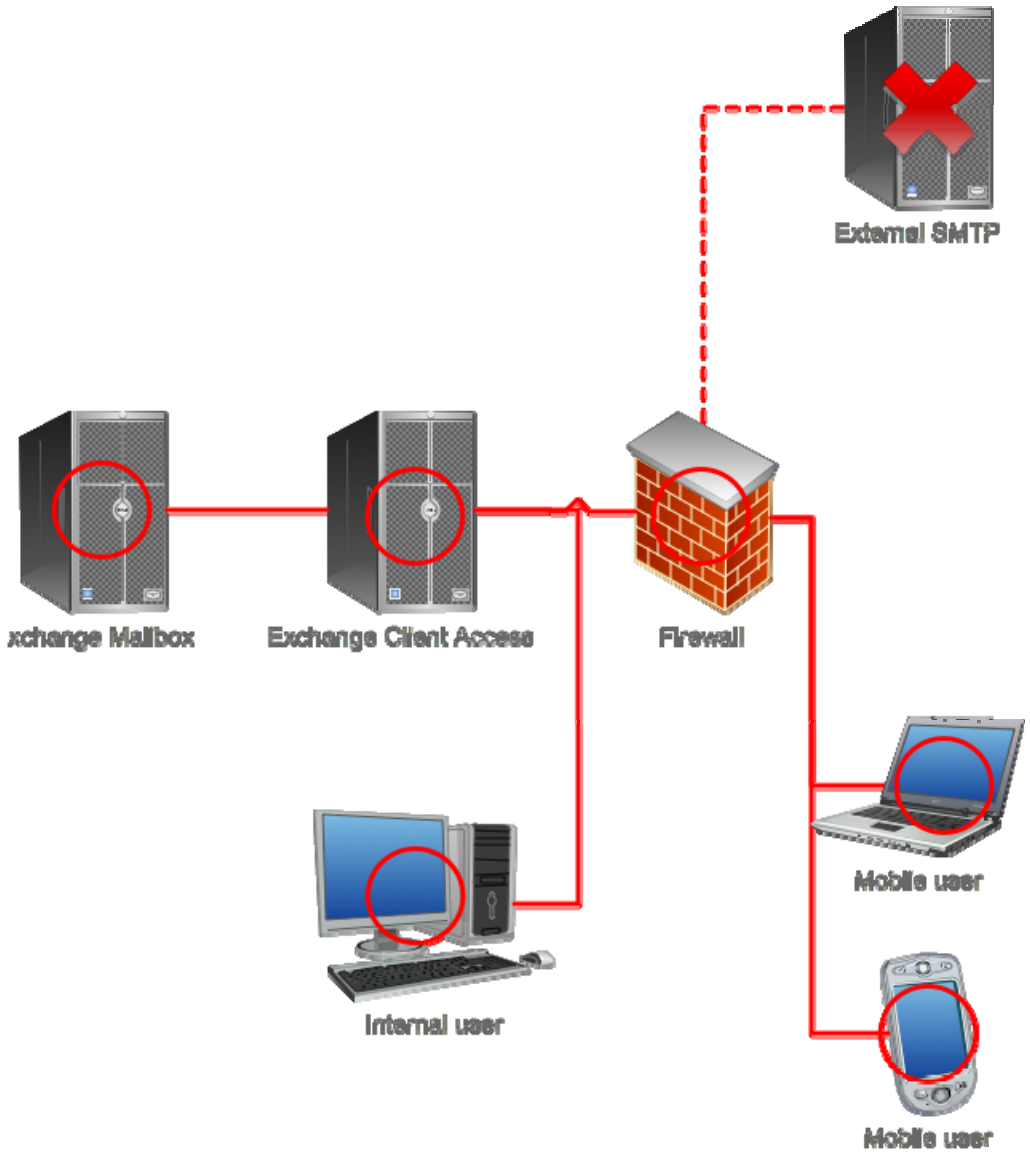

Figure 4.6: Identifying email security risks.

What does this tell us?

- Links to external SMTP servers, and their storage, can't necessarily be controlled and protected. We can't force the use of secure SMTP with external servers, and in any event, have no knowledge about how secure their storage is. That's a risk point, and the only way to mitigate it is through per-message encryption.
- All other links are under our control. We can secure them using SSL or other forms of encryption, and we can force that encryption to be used.
- The circles represent storage that's under our control. We can choose to use things such as drive encryption, remote wipe, and other tools to secure messages where they sit.

Now let's layer on instant messaging. We'll focus on Microsoft Lync Server and its official client applications, just to limit the possibilities that need to be depicted in Figure 4.7.

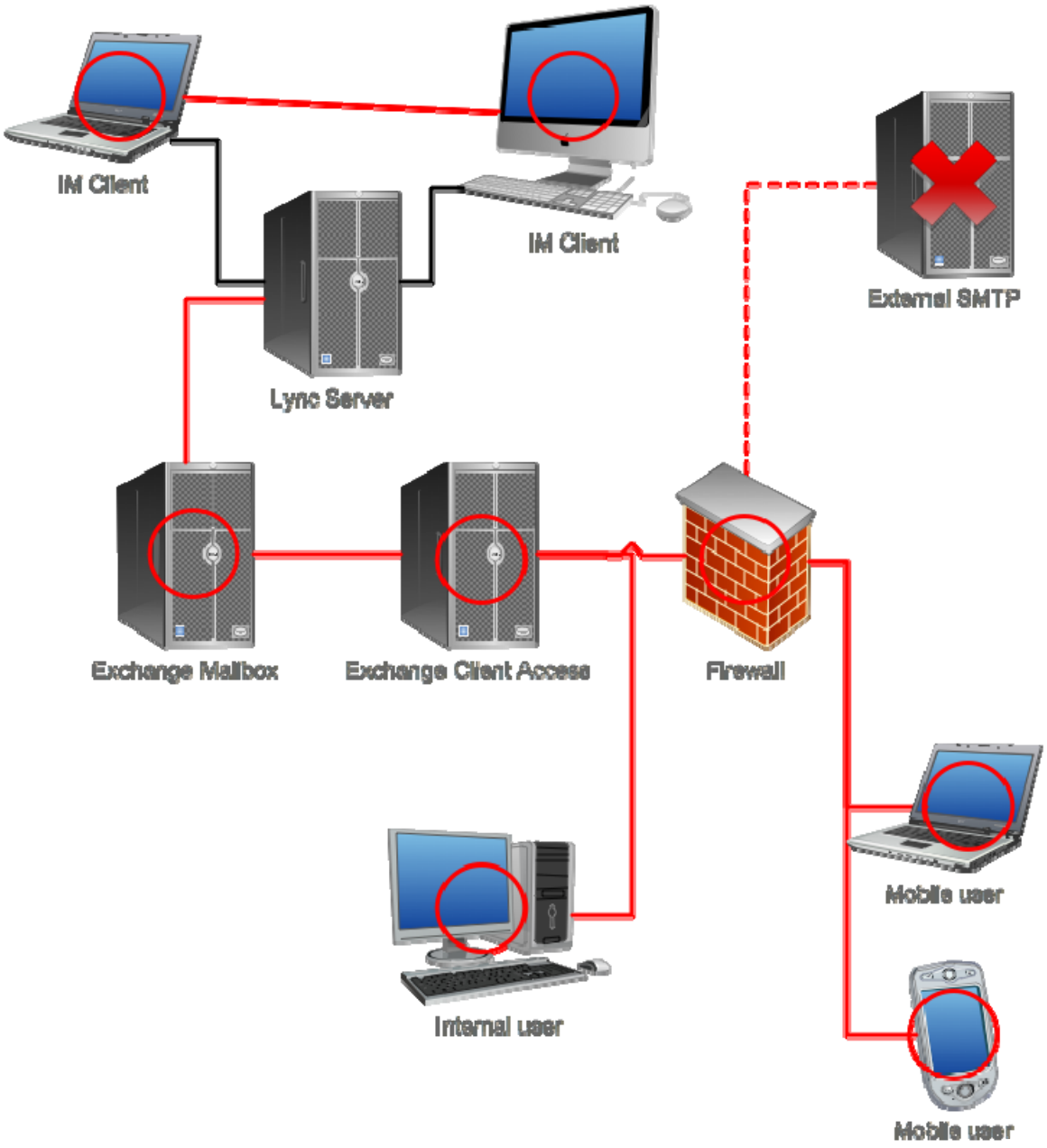

Figure 4.7: Considering IM security risks.

Whether the clients are internal or external, the communications with the Lync Server aren't necessarily a concern, as they typically provide only presence information. As we start using Lync Server to coordinate considerations such as meeting requests and dial-in, those links become a concern—and can be secured with SSL, just as with other Web-based communications.

A real concern is the traffic flowing between the two IM clients, and as already mentioned, *that* can be secured as well. IM clients often have options to log communications, meaning the storage where those logs are contained becomes a concern, where again full-drive encryption can offer a solution.

You simply keep repeating this process, layering on each additional communications component, until you've accounted for everything. Using MFT to move files from place to place instead of email attachments? See the illustration in Figure 4.8.

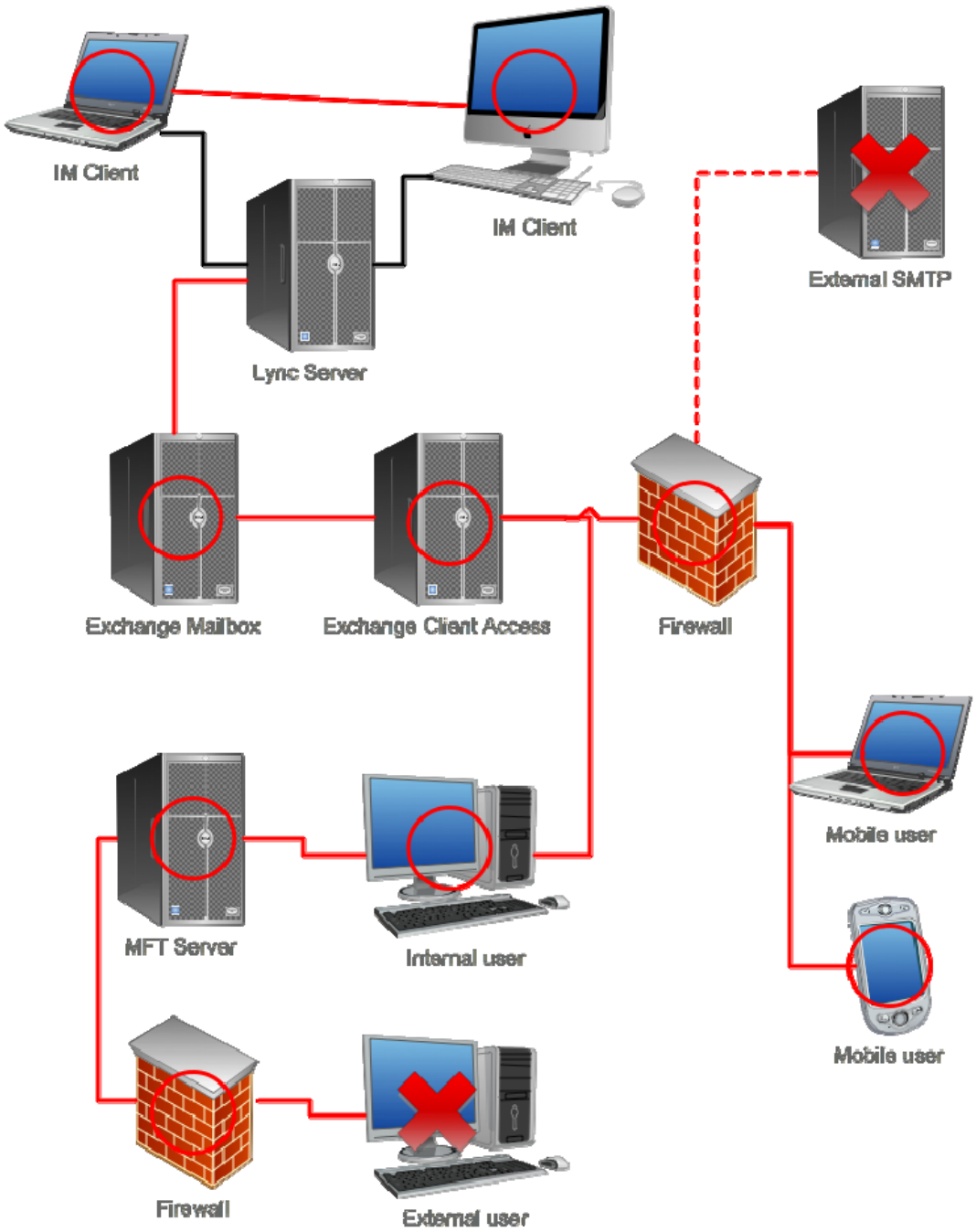

Figure 4.8: Layering on additional communications.

Here, we can clearly identify the links that we can *force* to be secured as well as the elements—such as the external user's local storage—that we have no control over. That indicates areas where per-communication encryption is appropriate, where we might simply have to rely on policies and business agreements to ensure privacy, or where privacy is simply beyond our reach. Provided you *know* where you can't be secure, and that you can *communicate* that limitation to the people to whom it matters, it's permissible to not be secure *everywhere*—especially because that's unavoidable anyway.

This kind of diagram can become an action plan. For example, if SSL will be used to secure communications, you now have a chart showing you which communications require it. You can then begin identifying actual communications endpoints, such as servers, and deciding what kinds of certificates you'll need to use. You'll be able to estimate how many certificates you need, start figuring out the pricing, and begin implementing.

The Social Channel: Making Users More Secure

The last—and, honestly, probably most important—thing is to really think about your users. I mentioned earlier that even users who are keenly aware of email security can be blissfully ignorant once they start using other modes of communication, especially ones—like VoIP—that resemble "harmless" modes that they're already familiar with (like telephone calls). Provide users with clear cues—like the additional visual cues a browser uses for EV certificates, as discussed in the previous chapter—and educate them to look for those cues and not bypass warning messages. The more they can be aware of what they're doing, the more effective your other security measures.

That said, pick your battles. Users can only keep track of so much, so try to focus on the things *you have no control over.* For example, don't train your users to type https:// when accessing OWA; configure OWA to auto-redirect to HTTPS if someone tries to use HTTP. That takes it out of users' hands, and means you don't have to worry about it. Focus on training them to make sure the Web browser's address bar says the right thing so that they know they're connected to *your* OWA—that's something only a user, paying attention, is going to see.

Summary

We've covered a lot of ground in these four chapters, and while this final installment was a kind of "everything else" collection, the overall effect will hopefully be to get you looking at your communications infrastructure in a new way.

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.