Realtime publishers

Tips and Tricks Guide™ To

Managed File Transfer

Don Jones

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

Volume 1

Each volume of this Tips and Tricks Guide will present a series of tips, tricks, answers, and best practices around Managed File Transfer.

Tip, Trick, Technique 1: When transferring files, isn't all encryption the same?

Definitely not. To begin with, there are numerous kinds of encryption—some of which can actually be broken quite easily. One of the earlier common forms of encryption (around 1996) relied on encryption keys that were 40 bits in length; surprisingly, many technologies and products continue to use this older, weaker form of encryption. Although there are nearly a trillion possible encryption keys using this form of encryption, relatively little computing power is needed to break the encryption—a modern home computer can do so in just a few days, and a powerful supercomputer can do so in a few minutes.

So all encryption is definitely not the same. That said, the field of cryptography has become incredibly complex and technical in the past few years, and it has become very difficult for businesspeople and even information technology professionals to fully understand the various differences. There are different encryption algorithms—DES, AES, and so forth—as well as encryption keys of differing lengths. Rather than try to become a cryptographic expert, your business would do well to look at higher-level performance standards.

One such standard comes under the US Federal Information Processing Standards. FIPS specifications are managed by the National Institute of Standards and Technology (NIST); FIPS 140-2 is the standard that specifically applies to data encryption, and it is managed by NIST's Computer Security Division. In fact, FIPS 140-2 is accepted by both the US and Canadian governments, and is used by almost all US government agencies, including the National Security Agency (NSA), and by many foreign ones. Although not mandated for private commercial use, the general feeling in the industry is that "if it's good enough for the paranoid folks at the NSA, it's good enough for us too."

FIPS 140-2 specifies the encryption algorithms and key strengths that a cryptography package must support in order to become certified. The standard also specifies testing criteria, and FIPS 140-2 certified products are those products that have passed the specified tests. Vendors of cryptography products can submit their products to the FIPS Cryptographic Module Validation Program (CMVP), which validates that the product meets the FIPS specification. The validation program is administered by NIST-certified independent labs, which not only examine the source code of the product but also its design documents and related materials—before subjecting the product to a battery of confirmation tests.

In fact, there's another facet—in addition to encryption algorithm and key strength—that further demonstrates how all encryption isn't the same: back doors. Encryption is implemented by computer programs, and those programs are written by human beings—who sometimes can't resist including an "Easter egg," back door, or other surprise in the code. These additions can weaken the strength of security-related code by making it easier to recover encryption keys, crack encryption, and so forth. Part of the CMVP process is an examination of the program source code to ensure that no such back doors exist in the code—further validating the strength *and security* of the encryption technology.

So the practical upshot is this: All encryption is not the same, and rather than become an expert on encryption, you should simply look for products that have earned FIPS 140-2 certification. Doing so ensures that you're getting the "best of breed" for modern cryptography practices, and that you're avoiding back doors, Easter eggs, and other unwanted inclusions in the code.

You can go a bit further. Cryptographic modules are certified by FIPS 140-2, but the encryption algorithms themselves can be certified by FIPS 197 (Advanced Encryption Standard), FIPS 180 (SHA-1 and HMAC-SHA-1 algorithms). By selecting a product that utilizes certified cryptography, you're assured of getting the most powerful, most secure encryption currently available.

Tip, Trick, Technique 2: Why do I need to manage person-to-person file transfers?

I used to work for an online retailer (who shall remain anonymous—they're no longer in business as an independent entity, in any event). Several times each day, we would transmit order information to a huge variety of vendors, who would process those orders and drop-ship them directly to our customers. Obviously, as we were transmitting a good deal of customer information, we were concerned about making sure that information stayed secure. We bought and built various components to enable us to perform secured file transfers, delivery confirmation, and so forth, and we required our vendors to explain how they secured our customers' data on their end. We tried, in other words, to do our best to keep that data confidential.

Imagine our surprise, then, when several of our customers complained that their personal information—including names and addresses—had been released, by us, without the customers' consent. Not surprised—shocked. We spent days poring over file transfer logs, pointing fingers at our vendors, and more—without finding a single point where that data could have been compromised. The answer finally came out in an all-company meeting where we explained what had happened, what we were trying to do about it, and so forth.

It turns out that some of our vendors would occasionally get a corrupted data file. Rather than call our technology department and request a re-transfer, the vendors were calling the customer service line, requesting order information for specific orders. Because we were a small company, our service representatives were familiar with the vendors, so they pulled the requested orders and simply emailed the information to someone at the vendor. The email, of course, wasn't encrypted in any way, and it in fact traversed several email servers between us and the vendor. One of those email servers was the point at which the data was copied and compromised. Oops.

This situation illustrates why you need to *manage* person-to-person file transfers. The employees in your company *will* engage in person-to-person file transfer. You can't stop them. If you prohibit email attachments, they'll use Gmail or Yahoo or some other alternative. Block access to those sites, and they'll start using sites like Drop.io or other "drop box" services. Block those, and your people will find something else. They *need* to be able to transfer files from person to person; failing an officially-supported solution, they'll find something unofficial. So you pretty much *have* to implement an official, supported, *secure* person-to-person file transfer mechanism that your employees—and your external partners—can utilize. What do those solutions look like?

Some look and work a lot like the peer-to-peer file transfer solutions that many consumers use at home. The person on either end of the transfer uses a file transfer client to send and receive files to and from each other. Typically, in a managed solution, the file data actually passes through a central file transfer server—in many cases, the same Managed File Transfer (MFT) server that is handling your server-to-server transfers, like the secure vendor transfers I described earlier. Figure 1.1 shows how this works.

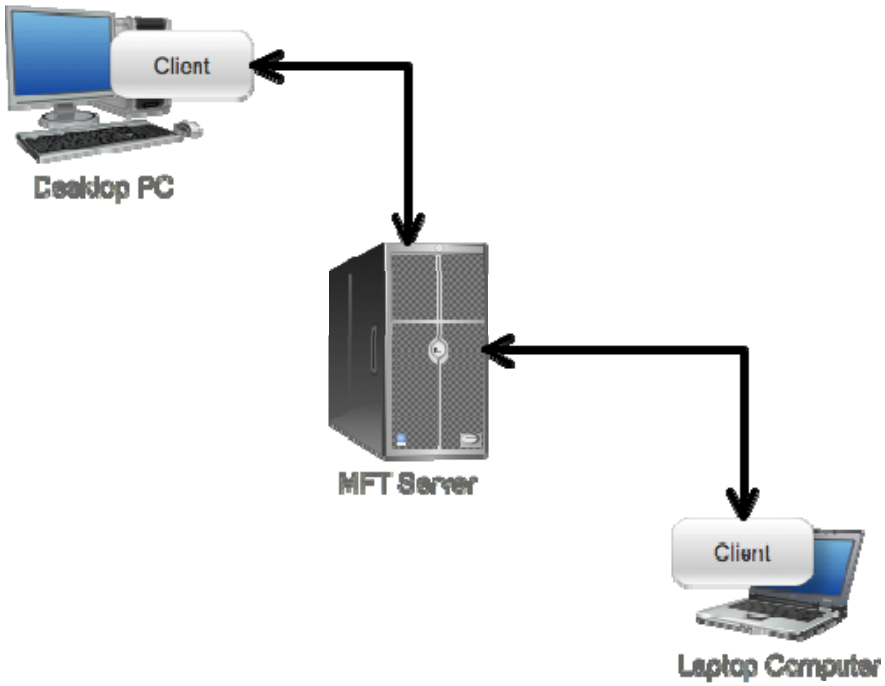

Figure 1.1: Person-to-person file transfer using an MFT server.

In this case, the transfer might be between an employee and an external partner, or even between two employees. In fact, this type of file transfer can help take some of the burden off your email servers (by eliminating large attachments), and can provide a more secure path for the data being transferred. That's especially important for companies dealing with legal and industry regulations that require the secure transfer of certain kinds of data.

Of course, the downside is that both parties must have a piece of client software, which is often not convenient for external partners. Many person-to-person solutions therefore offer alternative client interfaces, such as a Web site. This interface usually includes some kind of authorization, meaning you create either a temporary or permanent user account for an external partner. Figure 1.2 shows this alternative.

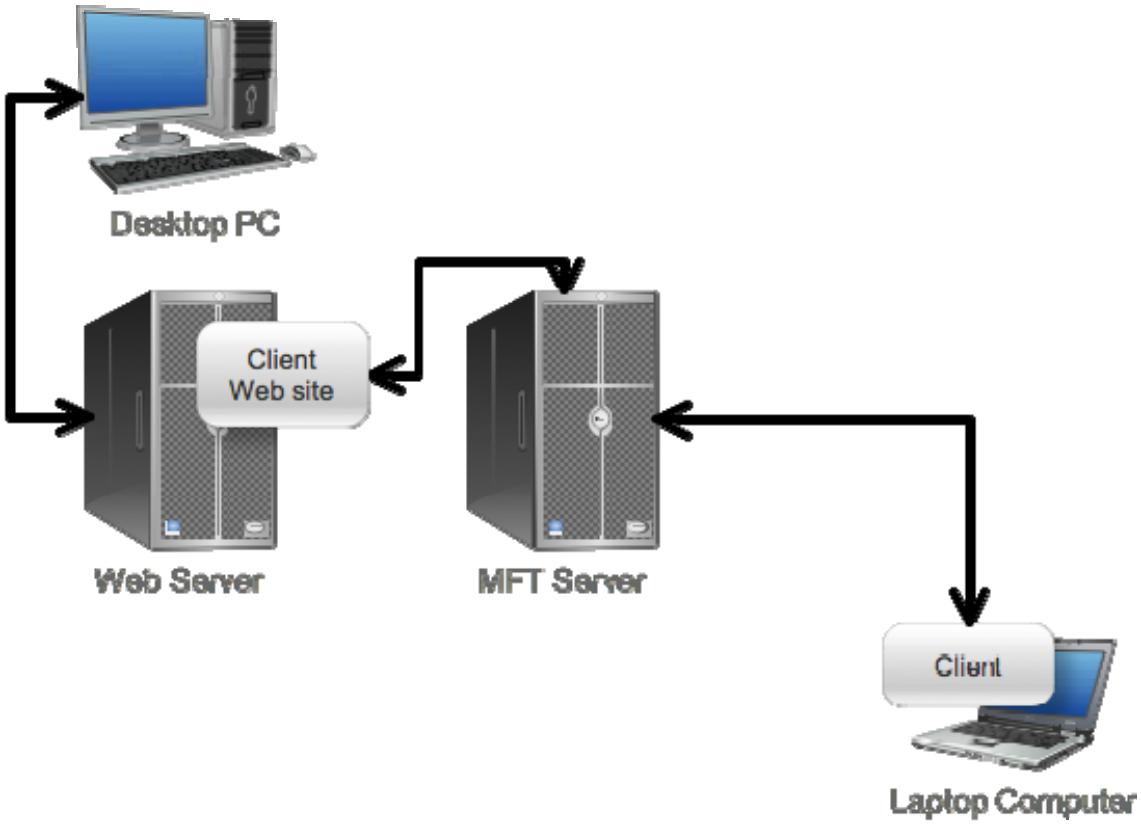

Figure 1.2: Using a Web interface for one file transfer user.

These person-to-person MFT systems often support asynchronous operation. That means the user sending the file can deposit it into the MFT system, then send a notification or "invite" to the intended recipient. The recipient often clicks a link to launch the client software or visit the secure Web site, authenticates, and then retrieves their file or files. Once the transfer completes, the sender might be notified of the successful transmission, and the MFT server might even be configured to automatically and securely delete the file from its own storage, preventing the file from "spreading" any further.

In practice, this type of system works very much like traditional email attachments, except that the "attachment" is being handled by a separate system that provides its own layers of security. There are a number of advantages:

- Unlike email, which can be intercepted by anyone, you're assured that only the authenticated recipient is receiving the file.
- Unlike email, which relies on a single transmission protocol that does not *mandate* encryption, the MFT server can ensure that an appropriate level of encryption is used.
- Email can traverse many intermediate servers, each of which can choose to retain a copy of the message and any attachments; MFT is more of a point-to-point transfer, and the MFT server can be configured to automatically remove any intermediate files if desired.
- Email systems typically contain little in the way of logging and tracking; MFT servers can often be configured to keep a detailed audit log of transfer activity.

This kind of person-to-person transfer doesn't need to be any more difficult for your users than sending email attachments—but it can be much more secure.

> **Note**
>
> It's even common for a person-to-person MFT solution to integrate with popular messaging clients, such as Microsoft Office Outlook, so that users can "attach" files to an email even more transparently. Such a plug-in might supplant the software's native file attachment functionality, and instead handle the attachment through the MFT solution.

Tip, Trick, Technique 3: Why do I need Managed File Transfer? Isn't FTP enough?

Possibly not. The Internet's venerable File Transfer Protocol (FTP) is usually supported by Managed File Transfer (MFT) systems, which can typically use FTP as one of the ways in which data is physically moved from place to place. However, MFT essentially wraps a significant management and automation layer around FTP. Consider some of the things an MFT solution might provide above and beyond FTP itself—even if FTP was, in fact, being used for the actual transfer of data:

- Most MFT solutions will offer a secure, encrypted variant of FTP as well as numerous other more-secure file transfer options. Remember that *FTP by itself doesn't offer any form of transport-level encryption* (although you could obviously encrypt the file data itself before sending, and decrypt it upon receipt; doing so involves logistical complications like sharing passwords or certificates).

- MFT solutions often provide guaranteed delivery, meaning they use file transfer protocols that give the sender a confirmation that the file was, in fact, correctly received by the recipient. This can be important in a number of business situations.
- MFT solutions can provide automation for transfers, automatically transferring files that are placed into a given folder, transferring files at a certain time of day, and so forth.
- MFT servers can also provide set-up and clean-up automation. For example, successfully-transferred files might be securely wiped from the MFT server's storage to help prevent unauthorized disclosure or additional transfers.
- MFT servers may provide application programming interfaces (APIs) that make file transfer easier to integrate into your internal line-of-business applications.
- MFT solutions commonly provide detailed audit logs of transfer activity, which can be useful for troubleshooting, security, compliance, and many other business purposes.
- Enterprise-class MFT solutions may provide options for automated failover and high availability, helping to ensure that your critical file transfers take place even in the event of certain kinds of software or hardware failures.

In short, FTP isn't a bad file transfer protocol—although it doesn't offer encryption. MFT isn't a file transfer protocol at all; it's a set of management services that wrap around file transfer protocols—like FTP, although that's not the only choice—to provide better security, manageability, accountability, and automation.

In today's business, FTP is rarely "enough." Aside from its general lack of security—which can be partially addressed by using protocols such as SFTP or FTPS instead—FTP simply lacks manageability, integration, and accountability. Many businesses feel that they simply need to "get a file from one place to another," but in reality they also need to

- Make sure the file isn't disclosed to anyone else
- *Ensure,* in a provable way, that the file got to its destination
- Get the file from, or deliver a file to, other business systems (integration)

In some cases, the business might even need to translate or transform a file before sending it or after receiving it. For example, a file received in XML format may need to be translated to several CSV files before being fed to other business systems or databases—and an MFT solution can provide the functionality needed to make that happen.

Many organizations tend to look at MFT first for its security capabilities, which often revolve around a few basic themes:

- Protecting data in-transit (encryption)
- Ensuring that only authorized individuals can access the MFT system (authorization and authentication)
- Tracking transfer activity (auditing)
- Reducing the spread of data (securely wiping temporary files after transfers are complete, and controlling the number of times a file can be transferred)

These are all things that a simple FTP server can't provide. Having satisfied their security requirements, organizations then begin to take advantage of the manageability capabilities of MFT systems, including centralized control, tracking, automation, and so forth—again, features that an FTP server alone simply can't give you.

Tip, Trick, Technique 4: How does Managed File Transfer help me meet and maintain compliance requirements?

Today's companies are dealing with an increasing array of legislative and industry requirements, mostly revolving around security. Legislation such as the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act (SOX), the Gramm-Leach-Bliley (Act) GLBA, the Payment Card Industry Data Security Standards (PCI DSS), Basel II, and more all have stringent data security requirements for specific types of data within your business—often the data that forms the core of your business, such as customer information or financial data.

Sometimes, these requirements are very technically precise. PCI DSS, for example, provides specific guidelines on what kind of data must be protected (customer and cardholder information), when it must be protected (in transit and when stored), and how it must be protected (encryption, in most cases). Other times, requirements are much more general and less technical in nature. HIPAA, for example, simply has a general requirement that patient information must not be disclosed to unauthorized parties; a 2009 addition to HIPAA also requires that data holders notify individuals when their protected information has been improperly disclosed.

Those general business-level requirements can be extremely difficult to implement from a technology perspective. For example, suppose you work in the healthcare industry and are subject to HIPAA. You need to transfer certain patient information to a partner company, and you need to do so in a way that complies with HIPAA. That means you need to actually implement several technical controls:

- Encrypt the data while it is in storage
- Potentially encrypt the data during transmission within your company, especially if such transmission occurs over a publicly-accessible network (such as when accessed by telecommuting employees)
- Encrypt the data during transmission to your partner
- Securely wipe any temporary copies of the data created during transmission
- Keep track of every access to the data while it is stored
- Keep track of every transmission of the data
- Store that tracking information in a secure, tamperproof database or log
- Control who can initiate transfers of specific kinds of data

A Managed File Transfer (MFT) system can help with many of these requirements. By using a properly-configured MFT system as your *sole means* of transmitting data—potentially both within your organization and externally—you can become compliant with these requirements much more easily.

An MFT solution—being primarily for *transfer* of data—obviously doesn't directly address requirements for the security of "data at rest"—that is, the data stored within your file servers, databases, and so forth. However, because MFT solutions often keep a temporary copy of any data being transferred, they *are* impacted by "data at rest" requirements. An effective MFT solution should fully secure access to such temporary files so that only the MFT system itself can access those files, and so that any access to those files is audited. Typically, MFT systems will rely on the underlying operating system (OS)—such as Windows or Linux—to provide the security and auditing for those files. A good MFT solution will provide the ability to automatically, and securely, wipe temporary files that are no longer needed, reducing the chance that those files will become the source of a data breach.

After the data goes into motion, the MFT solution's real value to your compliance posture kicks in. An MFT solution that has been certified to Federal Information Processing Standard (FIPS) 140-2 is automatically able to provide the level of encryption desired by most US- and Canadian-based security requirements; you simply have to ensure that your MFT system is configured to use a file transfer protocol that supports such encryption.

The MFT solution (a good one, at least) can track who has transferred a file, when a file was transferred, how long the transfer took, to where the file was transferred, what file was transferred, and so on. That information should be stored in a secure, tamperproof database that is not directly modifiable (except perhaps by highly-trusted administrators). A good MFT solution can also centrally control who can initiate transfers, and can use top-level management policies to govern what types of transfer protocols are used, what kind of logging is kept, and what types of files may be transferred.

The technologies used to bring about this level of compliance can be complex. In fact, simply providing the necessary cryptographic protocols can require an incredible amount of expertise, as a fully-compliant MFT solution will provide FIPS-validated cryptographic algorithms and modules. Obtaining that validation is expensive and time-consuming for a vendor, and requires a high level of software and cryptography expertise.

A thorough understanding of the compliance rules is also important. For example, some compliance efforts require you to notify customers of a data breach *only if the disclosed data was unencrypted.* That means applying encryption to the data, as well as to the transport stream, can help make your life easier: In the event that the data is improperly disclosed before or after transport, *it was still encrypted in and of itself,* so you haven't actually disclosed anything. As Figure 4.1 shows, MFT solutions can provide this functionality through an intermediate layer of file-based encryption, often using an industry-standard technology such as PGP.

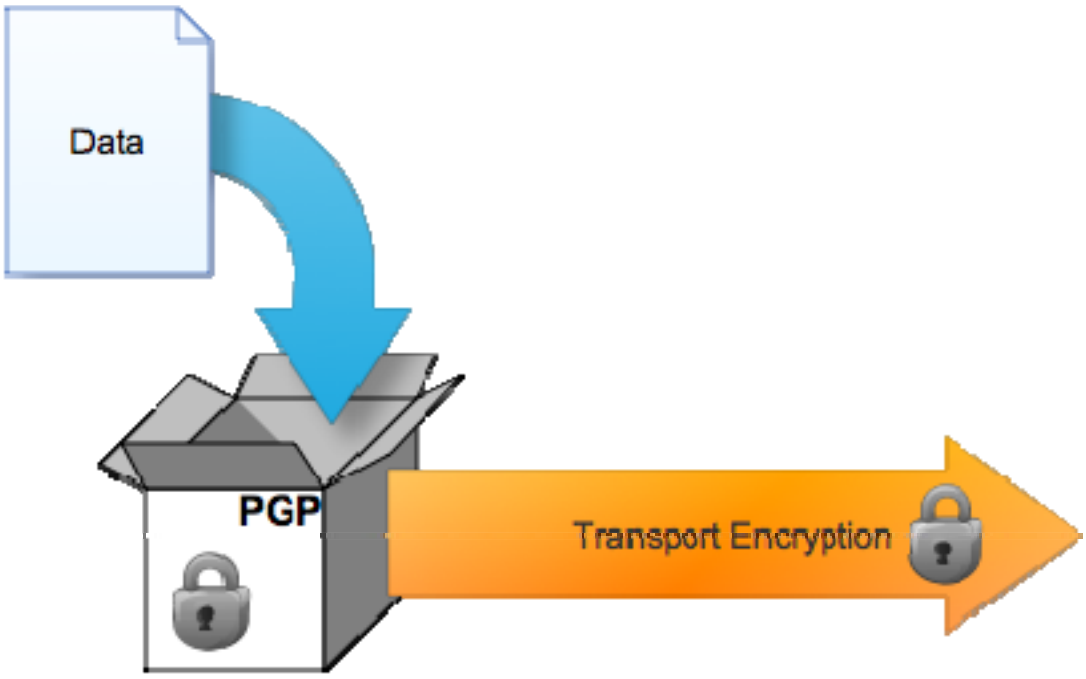

Figure 4.1: Multiple layers of encryption help meet different requirements.

By adding authentication to the mix—ensuring that both the sender and recipient verify each others' identities before beginning any transfer—you can further meet your compliance requirements. Using cryptographic hashes, such as the Secure Hash Algorithm (SHA), you can provide further protection by ensuring that the file isn't damaged or tampered with in-transit.

That's a *lot* of technical protocols and configurations that have to be set up. Building your own solution that supports SFTP, PHP, SSL, SHA, SSH, and a raft of other security-related acronyms simply isn't feasible. In fact, it's not even really practical for today's businesses to become experts in these fine-grained details as they evaluate MFT solutions. Let me explain: Commonly, businesses will attempt to translate compliance requirements into technical ones, then seek solutions that meet those technical requirements. I call this "double-mapping," and it looks something like the illustration in Figure 4.2.

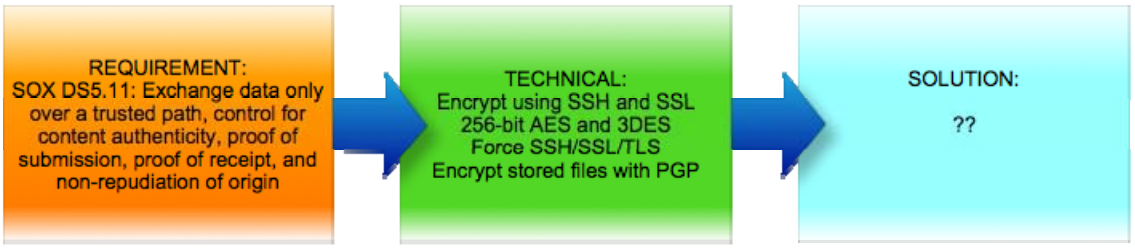

Figure 4.2: Mapping compliance requirements to technical ones, then to solutions.

This kind of effort on the part of businesses is no longer warranted. Compliance requirements have been around long enough to be well understood; vendors who are serious about providing compliance solutions should handle this mapping for you. Figure 4.3 shows what you should be looking for: A solution that maps its capabilities *directly to compliance requirements,* providing underlying technical explanations if you want them, but focusing on that business-level mapping.

Figure 4.3: Mapping solution capabilities to compliance requirements.

It's not that your business shouldn't be concerned about the underlying technical implementation; it's that you should be concerned *first* about *becoming compliant* and looking for solutions that map to specific compliance requirements—such as the SOX DS5.11 requirement in this example. Vendors can provide this through informative graphs and tables, such as the excerpt that Figure 4.4 shows.

DS5.11 Exchange of Sensitive Data. Exchange only over a trusted path, control for content authenticity, proof of submission, proof of receipt and non-repudiation of origin.	- User IDs and passwords always encrypted - Encrypts client connections over SSH and SSL (Implicit, Explicit and TLS) protocols - Session encryption using 256-bit AES encryption and 3DES - Force SSH, SSL/FTPS or TLS 1.0 or higher on all client connections to WS_FTP Server - Encrypts stored files with fully-integrated OpenPGP mode - Configurable SSL/TLS encryption down to the folder level - Policy based cryptographic strength enforcement - Import, export and create SSL x.509v3 certificates - Import, export and create SSH keys - Create, Edit, Import, Export, Delete OpenPGP keys with support for PGP, OpenPGP and GPG - Select and prioritize ciphers to use in OpenPGP key creation - Support for RSA and Diffie-Hellman key types with settable expiration date - OpenPGP asymmetric key length of 1024 – 4096 bits

Figure 4.4: Demonstrating compliance mapping.

This is truly the answer to the question, "How does Managed File Transfer help me meet and maintain compliance requirements?" You should be able to specify the legislative or industry rules that concern your business, and see exactly how a given solution addresses the specifics of those rules.

Tip, Trick, Technique 5: How can I ensure that my users utilize a Managed P2P File Transfer System?

I once worked for a company that spent tens of thousands of dollars implementing an electronic fax system. Incoming faxes were centrally received, scanned for recipient information, and routed via email to the proper recipient. Faxes could be sent by simply sending a properly-formatted email to a special email address. The system worked wonderfully and had very few problems. Nobody used it. They were all accustomed to printing pieces of paper, walking to the printer, and using the adjacent old-school paper fax machine.

Companies are looking hard at Managed File Transfer (MFT) systems to handle person-to-person (P2P) and person-to-server file transfers as well as server-to-server file transfers. MFT offers better security, better accountability, and better manageability than the existing mechanisms—likely email and instant messaging clients—that your users are probably accustomed to. The big question with any MFT P2P implementation, though, is "Will my users actually use it?" You can ensure a "Yes, they will" answer through two approaches: the carrot and the stick.

Users will employ whatever file transfer method they are most comfortable with and that is most convenient—that's why email and instant messaging clients are so popular for file transfers. They're easy, they're something users are using anyway, and they're something users have already learned how to use. The "stick" approach is to simply make these methods less convenient or completely unavailable:

- Block access to file-sharing Web sites
- Block access to Web-based email sites such as Gmail, Hotmail, and Yahoo
- Block users' ability to use FTP protocols from their client computers; you might simply block this at your corporate Internet firewall, for example, or you might block it at your internal routers to reduce internal use of unsecured, unmanaged FTP
- Curtail or cut off the use of file attachments to emails—either to external email recipients or to internal and external recipients, depending upon your requirements
- Filter file attachment traffic from instant messaging clients or restrict instant messaging to those clients that can be centrally prevented from allowing file transfers

The fact is that users *will find a way* no matter what restrictions and blockages you put in their way, but that's okay—you're not trying to block everything, you're simply trying to make these less-secured, less-manageable methods harder and less convenient to use. You'll also be offering a carrot: A more convenient, easier-to-use method that is official, supported, allowed, *and* properly securable and manageable. That's your MFT P2P solution.

Selecting the right solution is the key to having a tasty carrot to offer your users, and the "right solution" will depend largely on how your users transfer files today. If your users rely primarily on email attachments, look for a solution that integrates with your messaging client. This may be in the form of an alternate "attachment" button in the message composition window, for example. Rather than attaching a message to an email, this alternate button submits a file to your MFT server, and provides a link within the email for the recipient to click. It may even enable the sending user to create a temporary user account in the MFT system that the recipient can use to access the file—and may even send a separate notification email to the recipient with their new, temporary login credentials. Figure 5.1 illustrates this concept, which enables users to continue using what they perceive as "file attachments," but actually handles the file transfer through the MFT solution.

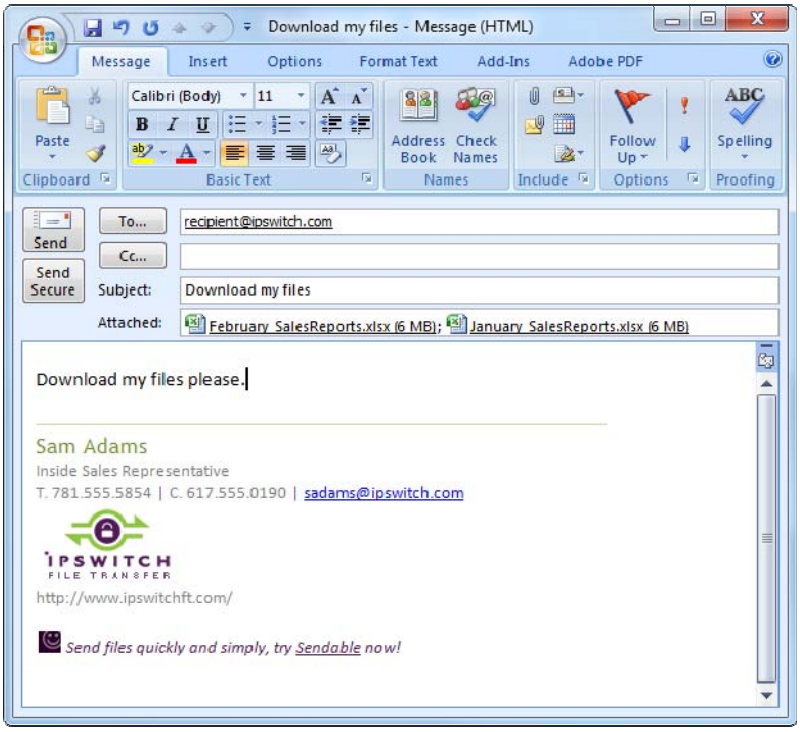

Figure 5.1: Sending e-mails "with attachments" through an MFT system.

If your users are accustomed to using FTP client software, look for an MFT solution that provides an alternative client that routes files through the MFT solution rather than connecting an FTP client directly to an FTP server. Your users experience should be very similar to using an FTP client—in fact, it may be better, because users can simply "submit" a file to the MFT server, then let it handle the actual transfer, any retries that are needed, and so on. The client should let them check the status of their ongoing transfer or receive a notification when it's completed.

If your users already use "drop box" Web services, an MFT solution should be able to provide you with a more secure, more manageable alternative. Web-based interfaces are increasingly common in high-end MFT solutions, and provide users with an easy, convenient way to submit files that can later be picked up by other internal or external users. In fact, the interface may be as simple as the one shown in Figure 5.2, where a sender uses a Web-based user interface to submit two files intended for delivery to an external user. The interface is similar to an email—something most users are very comfortable with, making this option both convenient and approachable.

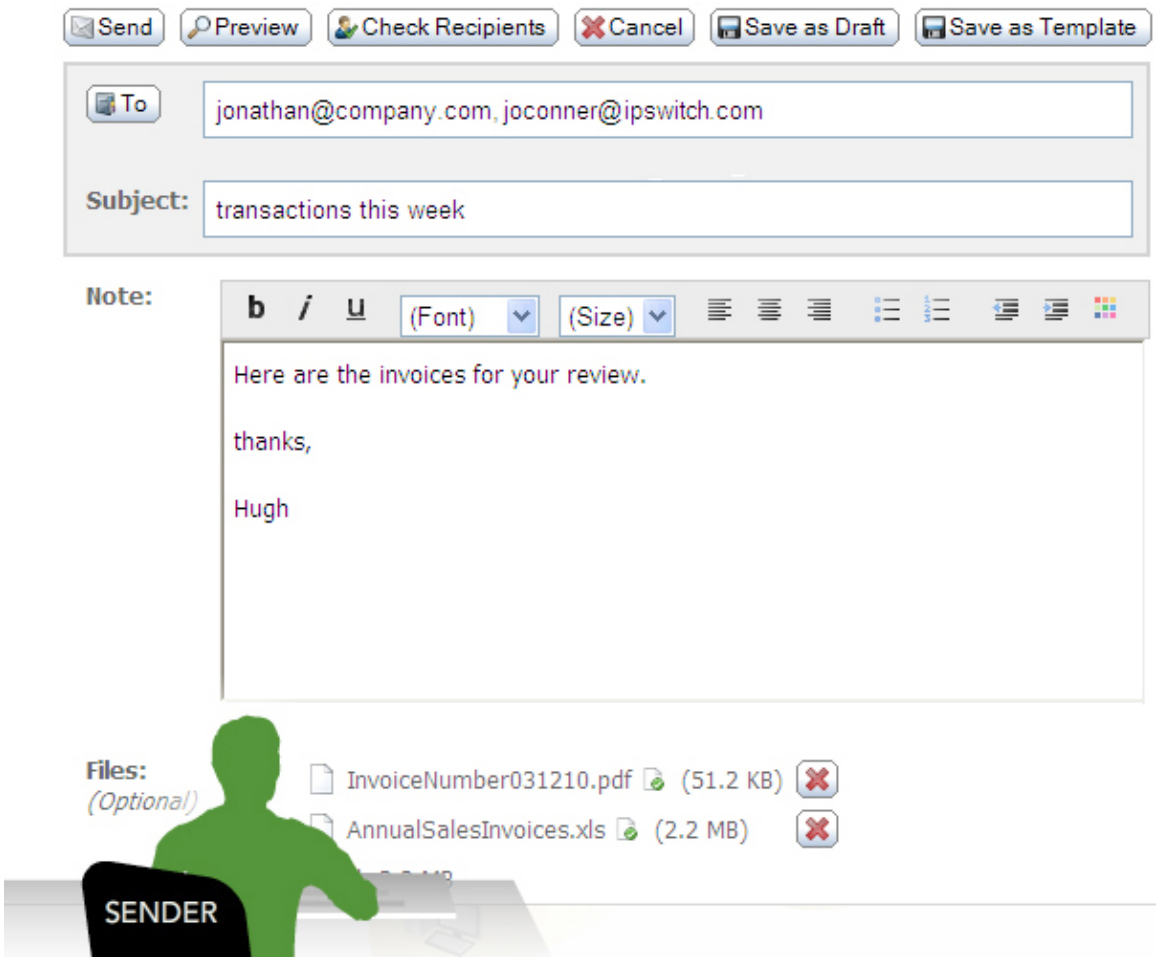

Figure 5.2: Using a Web interface to send files.

Other types of Web interfaces can make it easier for external users to upload files for internal recipients or to handle other types of ad-hoc, P2P file transfers.

The trick is to use both the carrot *and* the stick. Provide easy, convenient, understandable alternatives that work as much as possible like your users' existing methods for file transfer. Then make those existing methods less available, less convenient, and less easy. With convenient, supported alternatives in place, users won't look for unofficial methods. True, users will have to adjust their usage patterns somewhat either way—but the better an experience your MFT solution can provide, the more readily users will make the switch.

Before you make the switch, though, be sure you've prepared simple, visual materials to help users understand how these new alternatives are operated. Also make sure you've done a thorough inventory of use cases, and provided an alternative for each one—or at least deliberately decided that specific scenarios will no longer be permitted (be sure to document those and explain why, too—as well as offer suggestions on permitted scenarios that users may want to look at instead).

For example, you might decide that your users will submit files to your MFT solution, which will send an email notification to external users, advising them to come "pick up" their files. To ensure that only the proper recipient gets those files, you may decide to require authentication for those external users. If so, who will create the user accounts in the MFT system? If such a request needs to go through a Help desk, for example, the entire process becomes much less convenient than simply sending an email with an attachment—and your users will resist your new alternative. If, however, a temporary user account can be created ad-hoc and details can be sent to the recipient in a separate email, the process remains ad-hoc and convenient for your users—although you'll need to decide whether that approach satisfies your security requirements.

Tip, Trick, Technique 6: What kind of logging will I need for file transfers?

Aside from the obvious benefits of encryption and automation, logging (or auditing, if you prefer) is one of the biggest reasons companies choose to implement a Managed File Transfer (MFT) system. But what kind of logging capabilities can you expect from an MFT solution, and what should you look for?

First, as I explained in Tip 4, focus on your business requirements. What kind of logging or auditing do you *need?* Specifically in Tip 4 I suggested that MFT vendors should help you understand how their products help meet the specific conditions of whatever legislative or industry requirements you may be subject to, without necessitating that your business take the intermediate step of translating those requirements into technical capabilities first. If your logging or auditing needs come primarily from compliance requirements—as is often the case these days—you should simply be able to state your compliance requirements, and have the vendor demonstrate how their solution helps meet those requirements with regard to logging or auditing.

Many companies are incorporating external compliance requirements with their own internal needs. When considering what kind of logging you'll need, start with those internal and external needs as drivers. That said, there are a few general logging capabilities that most business will want:

- **A tamperproof audit trail.** Because nothing in technology can ever be "proof" against anything, you may see this positioned as *tamper-evident* instead, meaning that if tampering occurs, you'll at least know about it. This functionality is often implemented as a secure, and potentially encrypted, database in which the MFT solution stores records of transfer activity.
- **Log accessibility.** You'll need to report on transfer activity, and that means you'll need access to the log. Some MFT solutions will offer robust, detailed, built-in reporting, and others will provide direct read-only access to the database for use by third-party reporting tools. Ideally, you want both capabilities: Built-in reports get you up and running quickly, while database connectivity offers the ability to use external reporting, billing, and tracking applications.

- **Who.** Who transferred the file and who received it? The first who is often easy to determine for outgoing transfers, but you'll need an MFT solution that supports authentication, as well as non-repudiation, to log the identity of external senders and recipients. *Non-repudiation* is a feature that proves a file was received, and proves that it was not altered or corrupted while in transit. *Who* should also account for the physical servers involved, which can be authenticated through transport-level protocols such as SSL.
- **What.** Obviously, you'll want to know what was transferred. Typically, MFT solutions won't keep a copy of the file (although many can be configured to do so if you need an audit log that detailed, keeping a copy of the file can present additional security concerns), but they will log what file was transferred.
- **When.** When was the file transferred? When was the transfer complete? With large files, the difference between starting and completing the transfer can be significant—and you'll want to know both.
- **Where.** Where did the file come from? Where did it go? MFT solutions can often incorporate multi-step workflows that take a file through several steps, potentially involving transformation or translation, and each step should be clearly logged in the audit trail.
- **How.** How was the file transferred? What file transfer protocols were used? What encryption, if any, was used?

About the only thing you can't expect an audit log to include is *why* the transfer was made—although if you track the *who,* they can hopefully answer that question, if needed.

But there's more that you should expect in the log: errors. File transfers aren't always smooth and problem-free, and a detailed activity and error log can make it easier for administrators to troubleshoot problems.

Figure 6.1 shows what a basic activity log might look like. In this example, several files were transferred as part of an overall "task," and the activity log enables the task to be broken into each discrete operation—in this case, a series of five file uploads.

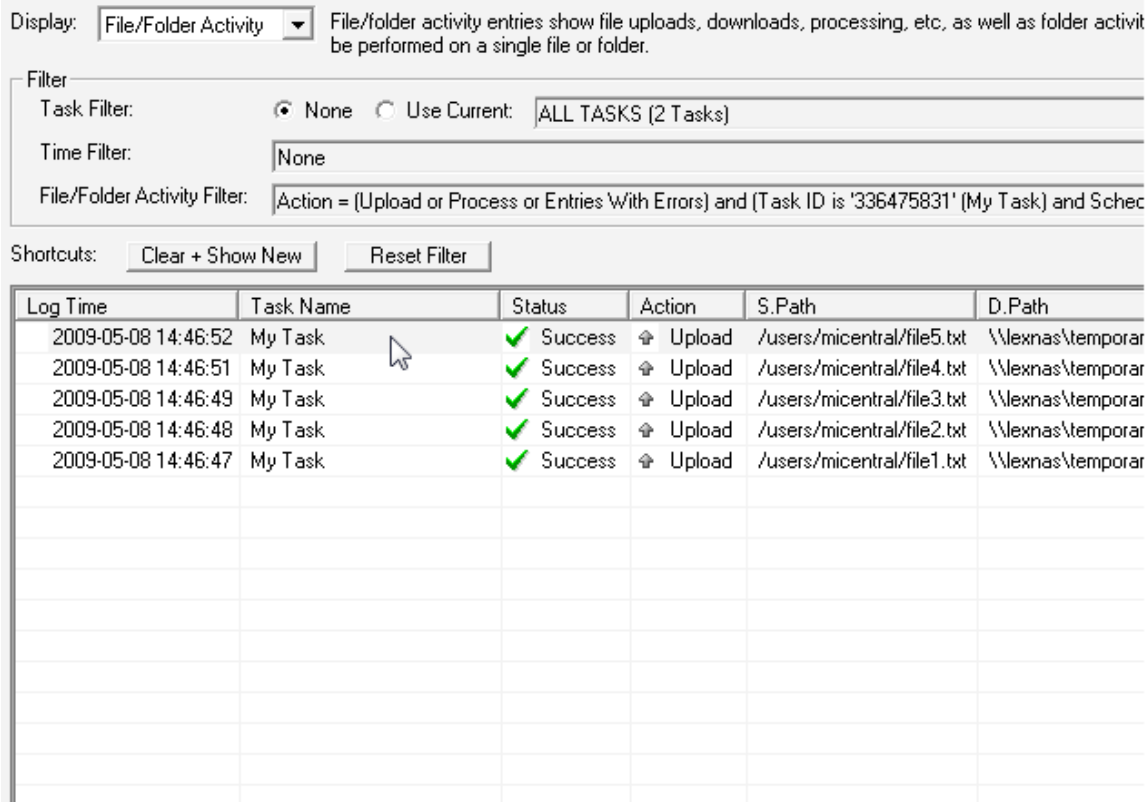

Figure 6.1: Example activity log.

The "tamperproof" or "tamper evident" nature of the log is perhaps one of the most important, and the most easily-overlooked. Particularly for companies who will rely on this log as a formal audit trail, potentially used to prove legislative or industry compliance or to use to track compliance breaches, this log *must* be trustworthy. MFT solutions will often rely on an internal database or on a major relational database management system (such as Microsoft SQL Server, Oracle, and so forth) to provide the necessary level of access control, encryption, and reliability.

Tip, Trick, Technique 7: Can a file transfer system enable central management and control?

You might say that the entire point of a Managed File Transfer (MFT) system is to do exactly that: provide *centralized* management and control. For example, let's say that your company is subject to the Payment Card Industry Data Security Standard (PCI DSS). Requirement 4 of PCI DSS is to "encrypt transmission of cardholder data and sensitive information across public networks," such as the Internet. Let's also say that you frequently need to transmit cardholder data to partner companies, such as vendors who will be fulfilling requests.

One option is to simply allow someone within your company to email that information, or to have an automated process do so. You'll need to ensure that everyone remembers to encrypt those emails—you did remember to get digital certificates for everyone, correct?—every single time. If someone forgets, you've created the potential for a data breach, and it's not going to look very good for your company on the evening news.

Another option is to automate the file transfer using an MFT solution. That solution can be *centrally configured* to always apply PGP-based encryption to the file, to *always* require an FTP-over-SSL connection with the vendors' FTP servers, and to *always* require 256-bit AES encryption. You don't have to remember those details beyond the initial configuration—it's centrally configured. Even if your users need to manually transfer something ad-hoc—perhaps an additional emergency order during the Christmas rush—your MFT solution will "know the rules" and act accordingly. Your users' lives become easier, your data stays protected, and everyone sleeps more soundly at night. This central control is often referred to as *policy-based configuration* because it's typically configured in one spot and enforced—not just applied—to your entire MFT infrastructure, regardless of how many physical servers and clients you are running.

What's the difference between *enforced* and *applied?* Making a configuration change is *applying* it. That doesn't, of course, stop someone else from coming along behind you and applying a new configuration. The idea with policies is that they're configured sort of on their own, and that they're protected by a unique set of permissions that govern who can modify them—they're not just wide-open to the day-to-day administrators who maintain your servers. In many cases, a review/approve workflow may have to be followed to make a change to a policy. Once set, the policies are *continually* applied to manageable elements such as MFT client software and MFT servers. A server administrator can't just re-configure a server, because the policy prevents it. The MFT solution ensures that your entire MFT infrastructure stays *properly* configured *all* the time.