A LINOMA SOFTWARE WHITE PAPER:

Beyond FTP: Securing and Managing File Transfers

EXECUTIVE SUMMARY:

Every day, millions of files are exchanged all over the world by corporations, government entities and other organizations. These electronic transfers include the critical data needed to conduct business, such as customer and order information, EDI documents, financial data, payment information, as well as employee and health-related information.

Most file transfers use a popular protocol known as FTP. This is a very aged protocol, since it was designed and implemented in the infancy of computing networks; even before the Internet was even heard of.

Few managers realize the security and management risks that have blossomed in their organization with the prevalent use of FTP. Fewer still have begun to take measures to bring the use of FTP into compliance with regulations such as PCI, SOX, HIPAA, State Privacy Laws or other mandates.

Few managers realize the security and management risks that have blossomed in their organization with the prevalent use of FTP.

This paper examines how FTP has become the standard for business-to-business file transfers. It identifies the key pitfalls that face management using this open protocol. Finally, it demonstrates how IT can begin the process of bringing the implementations of FTP into a more modern, secure framework. This new framework can increase user productivity and enhance IT manageability, while decreasing security exposures and adhering to growing compliance requirements.

LINOMA SOFTWARE

How FTP Became the "Standard"

FTP stands for File Transfer Protocol and was one of the first formalized networked applications provided by TCP/IP. In fact, it began years before the TCP or IP protocols even existed. Its first appearance was among a list of requests proposed in 1971 for the ratification of the Network Control Protocol (NCP), a precursor of TCP.

The early FTP designs, created almost 40 years ago, look remarkably similar to what we use today. As various networking protocols superseded NCP (Arpanet, TCP, IP, and finally TCP/IP) the FTP specifications tagged along, evolving only slightly, while retaining its basic structure and functionality. And because it has not evolved, FTP is a protocol rife with security exposures and management issues.

Some file transfer scripts are so deeply embedded in applications that IT is only vaguely aware of their detailed functions.

Good News/Bad News

The good news is every computer has a built-in capability to exchange data: The file transfer application that enables FTP is embedded in the services of TCP/IP.

The bad news is – after nearly 40 years of casual implementation – many business-to-business FTP implementations are uncontrolled and insecure, robbing corporations of productivity and endangering the security of their critical data.

Today, FTP remains a critical lynchpin in business-to-business communication. It is the facility most often used in file transfers between diverse computing platforms. The manner in which FTP is commonly implemented by businesses today needs to be seriously readdressed.

Pitfalls of FTP implementations

Let's examine three basic pitfalls common with FTP deployments:

- User interaction
- Complexity
- Security

User Interaction

Many organizations allow their employees to perform file transfers directly from their desktops using PC-based tools. However, these tools require manual intervention and are error-prone (e.g. the user may accidently choose the wrong files to send).

Some file transfers may rely on more detailed steps to interact with a distant server, requiring knowledge of the underlying protocols and formats to utilize. As these transfers become repetitive or mission-critical, IT is often called upon to help automate those processes within their business applications.

As a tactic to simplify file transfers, FTP has a built-in ability to accept scripts (sometimes referred to as "batch files") that essentially send keystrokes to the FTP client program. This scripting function mimics the commands that would be hand-typed into the terminal interface, allowing for automation of FTP transfers.

Typically the skills of a programmer are needed to help build any required FTP scripts. These scripts often will contain commands to login to the server, get files, put files, remove files, log out, etc. But FTP scripts have their own severe limitations:

- Scripts have no logic-branching capability, so they can't be programmed to make real-time decisions based upon a server's response.
- Scripts can't accept prompted user input to guide the process of sending files.
- Scripts have no inherent means of notifying the user or application when an error has aborted the transfer.

So while FTP scripts circumvent the pitfalls of the manual FTP tools, their usefulness can be limited.

Complexity and Costs

As soon as FTP functions are obscured in program scripts and/or migrated off the centralized system, IT departments are faced with a number of challenges.

Configuration Changes

When the settings (e.g. host names, IP addresses, folder names) for FTP servers change, IT often has to devote resources to reconfigure FTP clients or maintain embedded scripts.

The same problem exists when User ID credentials and passwords change, or when new business partners are added.

If the underpinnings of FTP transactions are decentralized, IT personnel may have to go to mutliple locations to make these changes.

...if FTP transactions are embedded or decentralized, IT personnel are often scrambling to make any needed changes...

Legacy Support

How many times has your programming team discovered a FTP script inside a custom program or procedure? The person who wrote the application is long gone, but now the code must be maintained because the transfer of data to or from the business partner is crucial.

Specialized legacy implementations of FTP are ticking time bombs. They hamper the ability of IT to quickly respond to changes in the business model, and they leave important user credentials and passwords in an insecure state.

Migration to PC Issues

Any decision to distribute corporate FTP functions away from the central server and out to PCs should be carefully considered.

Decentralizing FTP will often introduce serious inefficiencies and increased personnel costs. Lag times and errors will also likely increase as more steps and devices need to be utilized "in the chain" to get the files in the right format and places at the right times.

However, when it comes to management problems, FTP's security exposure is the greatest danger facing the organization.

FTP Security Exposures

FTP was designed as an easy mechanism for exchanging files between computers at a time when networks were new and information security was an immature science. In the 1970s, if you wanted to secure a server from unwanted access, you simply locked the computer room door. User access to data was controlled by the basic User ID and password scenario. The Internet did not yet exist and the personal computer revolution was still a decade away.

Today, the security of business file transfers is of paramount importance. The exchange of business records between computing systems, between enterprises, and even across international borders has become critical to the global economy.

Yet, the original native FTP facility of TCP/IP wasn't designed for the requirements of the modern, globally connected enterprise. FTP's basic security mechanisms – the User ID and password -- have long ago been outdated by advances in network sleuthing technologies, hackers, malware, and the proliferation of millions of network-attached users.

So what are the security issues facing organizations today?

Typical implementations of FTP **do not meet** the requirements of today's compliance regulations.

Regulatory Compliance

Regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), State Privacy Laws, the Food and Drug Administration (FDA) 21 CFR Part 11, and Sarbanes-Oxley (SOX) – place significant requirements on companies regarding the exchange of data.

For instance:

- PCI requires that all credit card numbers be encrypted while "at rest" or "in motion". Failure to do so can result in severe fines and potential loss of your merchant account.
- HIPAA requires that companies demonstrate that only the intended information is shared or exchanged.
- The FDA requires that administrative controls are in place when electronic systems and records are used in place of paper or manual systems.
- Sarbanes-Oxley requires that business processes – which include automated file transfers – are auditable.
- State Privacy Laws require that customers are notified if their personal information may have been lost or stolen. Some states can assess large fines against organizations if this data is not protected properly.

Typical implementations of FTP don't meet the requirements of compliance regulations such as these.

Auditing

Conventional FTP does not natively maintain a record of file transfers. Business processes that rely on FTP to exchange information are simply not auditable using this basic facility. For instance, how do you know what files are being transmitted? Who are the files being transferred to?

PC FTP Applications

Too many organizations have moved the functionality of business-to-business file exchanges to personal computers. But moving data to PCs for these FTP functions can leave sensitive files vulnerable on those machines.

Building safeguards to ensure that no data is left in the open on a PC is a costly and error-prone process. Yet, unless IT implements specific security steps it's difficult to ensure that data sent through a personal computer has been adequately scrubbed from its hard-disk after the transfer.

IT needs to know how and when vital company information is leaving the main system, and how it is being used or transferred to other systems.

Business processes that use plain FTP for transferring data should be closely examined.

Script File Exposure

The use of FTP scripts or batch files leaves User IDs and passwords in the open, where they can easily be hacked.

Script files can expose not only your data to misuse, but can also expose your business partner's system to attacks.

IT should avoid the use of embedded scripts except for the most generic of file transfers where User IDs and passwords are not required.

File Encryption Processes

Native FTP does not encrypt data. Consequently, files are often transferred as "clear text", leaving them open to hacking.

Where file-level encryption is used, too often it requires a two-step process to first encrypt the data and then to send it. Furthermore, the Keys or Passwords used to encrypt and decrypt these files are often not secured and managed properly.

Best Practices for Secure File Transfers

Clearly, the manner by which businesses are using FTP needs to be reexamined and strengthened. But how should IT begin?

Research

The first step is to examine how FTP is being used in your organization.

- What kinds of data is being sent or retrieved over FTP?
- Where do the FTP client applications currently reside?
- What are the reasons for distributing FTP functions (if any) to personal computers or departmental servers?
- Where are FTP scripts being used?

The answers to these and other questions will help you understand the breadth of the security and management problems facing your organization with FTP.

The next step is to identify how the organization needs to manage and secure file transfers.

- What are the data exchange requirements of your organization and your trading partners (e.g. customers, vendors, banks)?
- What are the pertinent compliance regulations that must be met?
- What are the requirements for encryption and authentication?
- What are the realistic expectations from users who are responsible for the day-to-day transfers of data?

How can IT implement a solution that performs file transfers in a manageable and secure manner? There are a number of common elements in a successful solution.

Deploy a System-wide, Comprehensive, and Configurable Solution

In the past, IT had no system-wide approach to file exchange: stand-alone FTP tools and scripts was considered to be enough to get the job done.

But today, as business-to-business transfers proliferate, it's time for IT to deploy a strategy that meets the overall requirements of security, flexibility, and ease-of-use.

Here are some basic guidelines that can help IT devise this strategy.

Location, Location, Location

The best solution to securing your FTP implementations will be one that best centralizes and manages the control of those transfers. The practice of distributing file transfers off the main information system complicates management and opens security holes. How does centralizing FTP reduce the number of management issues?

The best solution for securing and controlling file transfers is to manage those functions from a centralized system.

Centralization:

- Maintains the rigor of the native operating system's security mechanisms.
- Sustains the compliance requirements that have been already been implemented on the host system. This includes authority controls and reporting prerequisites.
- Provides a single-point of maintenance for all FTP user profiles and passwords.
- Contains standardized data encryption techniques and centralized key management. Instead of building subsystems for encryption on individual user platforms, IT can engineer a comprehensive solution that provides better control and security.
- Provides a centralized logging system of all file transfer activity for auditing purposes, along with descriptive error logs and message alerts when transfers fail.

Coordinate Access to FTP Functions

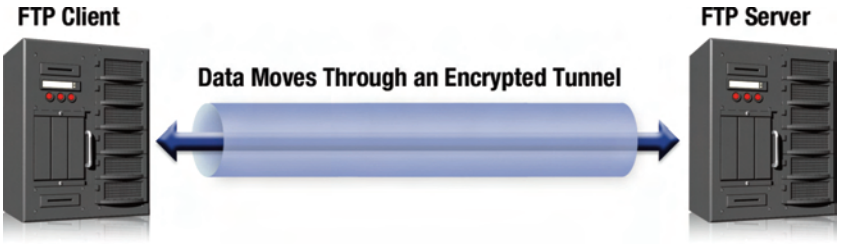
As we noted earlier, home-built or custom applications that nest FTP functions within application code are management time bombs: They obscure the configurations and activities of file transfers, and they can leave User IDs and passwords in the open.

If IT has applications such as these, they should utilize a centralized approach to access encryption and FTP functions through a controlled framework. This strategy will enable the control of server and user credentials, organize configurations, and permit the implementation of other management capabilities such as compliance logging for auditing.

FTP functions contained within application code are management time bombs.

Deploy Encryption

One of the best solutions for protecting your FTP transmissions is to upgrade to "Secure FTP" encryption technology. The two popular Secure FTP protocols are named SFTP (meaning FTP over SSH) and FTPS (meaning FTP over SSL). Both SFTP and FTPS will create encrypted tunnels between your system and your trading partners. In essence, anything that flows over those tunnels will be protected, including any user ids, passwords, commands, as well as any data that is transmitted.



One of the main differences between SFTP and FTPS is the way authentication is performed. With SFTP, clients can be authenticated with just a password or a Private Key. With FTPS, clients and servers can be authenticated with certificates, which are either self-signed (by your organization) or signed by a Certificate Authority (e.g. Verisign).

Choosing the right type of Secure FTP protocol to use will depend on your trading partner's capabilities and authentication requirements. You should not leave it up to your users to decide which secure protocol or methodology works best. This can create a hodgepodge of approaches, none of which may meet your overall security and authentication policies.

This is an area where IT's expertise is required to ensure that the right form of encryption is utilized, that authentication mechanisms are properly implemented and that regulatory requirements have been met.

Invest in Ease-of-Use User Paradigms

Finally, don't forget the End-users and their needs for automation and an easy-to-use interface.

By centralizing FTP functions, IT can provide a standard of usability that automates file transfers in a secure and productive manner.

But if the user's interface to the functions of FTP doesn't meet the ease-of-use expectations, the success of centralizing FTP can be threatened. In the end, users may reassert their demands for PC-based, distributed solutions.

Controlling the FTP Exposure

As a file transfer mechanism, FTP has had a long and illustrious history. The use of this standard protocol will remain a key facility for business as long as TCP/IP continues to define the protocols of the Internet.

But the past, ad hoc implementation of FTP tools throughout the enterprise, the past attempts to automate its functions, and the inherently weak security of its basic design has left IT with an infrastructure that does not meet the requirements of manageability or security. In most cases, the standard use of FTP by organizations is a security hazard waiting to become a corporate catastrophe.

IT must bring FTP into compliance, must secure the file transfer processes, and must reengineer how the facility is used and controlled within the organization.

The best solution brings FTP into a managed environment that controls and centralizes access, provides inherent encryption technology, and streamlines the entire process. With the right solution in hand – designed with centralized control in mind – IT can continue to automate the business-to-business exchange of crucial data, meet the requirements of compliance, and satisfy the need of its user base for flexibility and ease-of-use.

IT must bring FTP into compliance, must secure the file transfer processes, and must reengineer how the facility is used and controlled within the organization.

LINOMA SOFTWARE

About Linoma Software

Founded in 1994, Linoma Software provides innovative technologies to consistently meet evolving needs for managing, automating and securing data transmissions. Linoma Software has a diverse install base of over 3,000 customers around the world including Fortune 500 companies, non-profit organizations and government entities.

Website: www.GoAnywhereMFT.com

Toll-free: 800.949.4696

Outside USA: 402-944-4242

Email: sales@linoma.com