Global Information Assurance Certification Paper

Ron VanEtten

Version 1.4b

Title:

Mitigating the risks of using FTP for automated health and financial data transfers over the Internet.

Abstract:
The purpose of this paper is to show how a Fortune 100 Company (F100C) took advantage of the Internet to limit the growth of the leased lines used by automated processes to transfer data files. F100C also expressed a desire to replace the site to site data transfers that used tapes or CD's sent by courier. F100C business partners wanted to enjoy the popular belief that money could be saved by sending files over the Internet. The goal was to meet expectations while not compromising the security of the data or internal trusted networks. Guaranteeing the confidentiality of customer data has become even more important to meet compliance required by governmental regulations imposed by HIPAA and GLBA.

Business reason for using File Transfer Protocol (FTP):
FTP was the primary method of transfer already being used over private leased lines. It was assumed that the transition to the use of the public Internet would be simple. This would be the case, if the Internet was a secure network, but it is not. F100C technical support was in favor of the use of SSH/SCP for the transfers, but negotiations with F100C business, legal and purchasing partners showed that this was not feasible. Since the use of FTP was not negotiable, F100C technical support had to provide a means to protect data during transit between sites. Because FTP is ubiquitous, it is available to all of F100C as well as any vendor business partners who need to share data with F100C. The global availability of FTP removed the burden of for external vendor business partner training and support from F100C.

Reason for choosing Pretty Good Privacy (PGP®):
Since FTP is an insecure method for transporting data, F100C needed to provide a method to secure the data prior to transport by FTP. There are many reasons for choosing PGP® including: it uses private/public encryption keys; has good performance for the encryption decryption process; can be isolated from the FTP server; and transport of the encrypted data by FTP is identical to unencrypted transport. It is available for virtually all machine platforms and operating systems from desktop to mainframe. F100C's legacy applications were robust in handling FTP. The addition of PGP® encryption was a separate step that could be isolated and perfected without infecting the existing process with unknowns.

An important result of the choice of PGP® was that the payload data is encrypted before any transport of the data is attempted and decrypted after being successfully transferred. During the development of the data transfer processing, the data is not exposed to the public in the clear. Having the payload encrypted mitigates the risk of compromising the data if there is a problem with the connectivity between sites, or if the business partner changes its Internet address without notifying F100C.

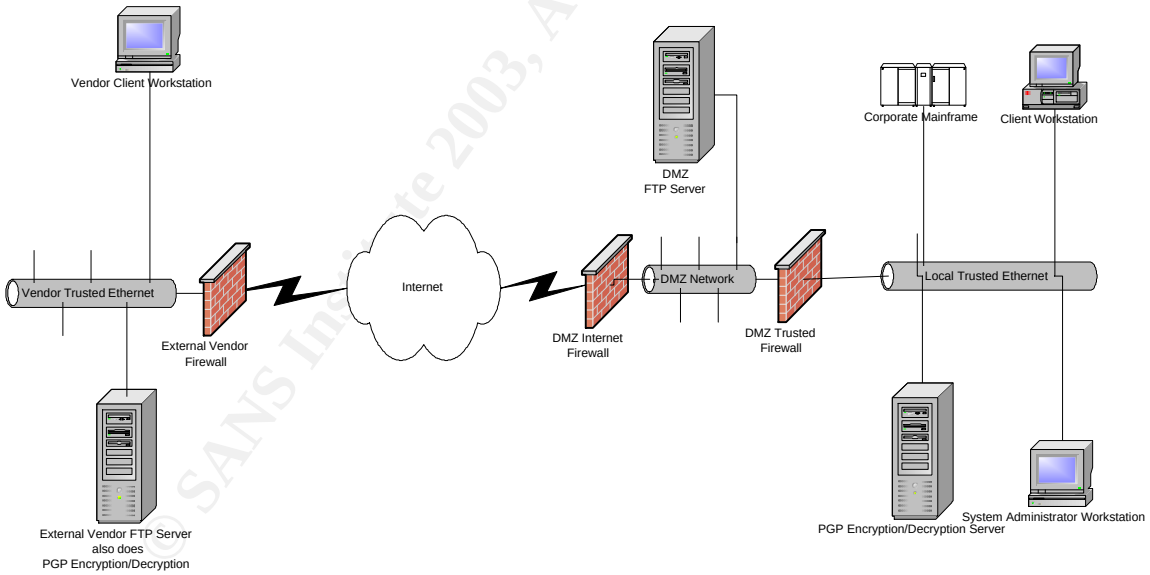
Use of PGP® also allows us to make the data only available for the intended recipient at the time of encryption by choosing which public keys to use to

encrypt the data. This provides a mitigating control on the file manipulation processing (discussed later) that is used on the FTP server to make files available for the intended vendor.

PGP® provides other advantages for distribution and maintenance. Since the PGP® process uses key pairs that are generated at each site, F100C was isolated from maintenance of the keys at the various vendor sites after initial training. It was a business decision to require that the "public" keys not be put on an Internet accessible key server. The "public" keys are emailed between the external vendor and F100C for use with this processing. The one time exposure of the "public" keys was considered an acceptable risk.

One difficulty experienced with the PGP® keys was differences between use of Diffie-Hellman encryption keys and RSA encryption keys. This paper is not about the intricacies of PGP® so suffice it to say, F100C discovered that the mainframe implementation of PGP® must be chosen before choosing any other implementation of PGP® due to limitations of the available versions of PGP® for the mainframe. Another problem with the main frame version of PGP® was configuring the record size and blocking of the records. Once these issues were resolved, PGP® proved to be very robust and reliable.

Network architecture:



Logical view of the Secure Transmission of Files VIA FTP

The goal was to mitigate as much of the risk as possible associated with the use of FTP. The file handling duties were split between servers in Trusted and DMZ security zones. The FTP server was put in the F100C Internet Demilitarized Zone (DMZ). In the trusted network, another server provides for encryption and decryption of the files with that server becoming a client to the FTP server in the

DMZ. Since this would be the case for all business partners who need access to the DMZ FTP server, F100C had a model that they could use to build any servers from.

Hardening the FTP server:
The service required for this Internet activity is File Transfer Protocol (FTP). As FTP is currently listed number five in the SANS Top Twenty vulnerabilities (SANS), server hardening is required[2]. F100C performed the following these well known procedures to harden the FTP server (Garfinkel p. 487).

1. All necessary patches were applied to the server. F100C used different server manufacturers based on the support that was available at each location. The primary contact for the current patch levels was on-site customer engineers from the respective server vendors. Additionally, vendor web sites were consulted for lists of the latest patches.
2. Services running on the server were minimized. This includes paying careful attention to the services available for inetd in /etc/inetd.conf.
3. The number of users on the server was minimized. Internal support staff was generally expected to have log on capability to all the servers that they may support. For this particular group of servers, every effort was made to limit the personnel listed in the password file to only the subset of those who would actually support the server. Caution is advised in reducing the list of support staff so as to still provide required support for the server.
4. Enable verbose logging for FTP. As this server was located in an untrusted zone, getting the most complete logging was considered critical to track down malicious and non-malicious activities on the server. The logs grow very large and need to be rotated daily and archived to a log server which follows F100C records retention policy.
5. Use anonymous FTP to lock external users into the chrooted FTP environment. Implementing FTP in this manner allows for the verification of connectivity by allowing the anonymous user to perform an ls of the directory, without giving them any additional rights on the server.
6. No other access is permitted to the user until they log in with an assigned ID and password. Once logged in, the users must switch to a real ID to get or put files.
7. Shadow the password file. This is to be expected when building any server.
8. Not allow FTP users to log on to the server and open a shell. Create a bogus shell that does not allow any FTP user to open a real shell. This requires a couple steps as shown below in which a dummy shell program is created, added to /etc/shells or /etc/security/login.cfg and entered as the shell the ftp users get in the passwd file.

9. Limit the user accounts that can have access to the FTP application. Add entries to the /etc/ftpusers file so that they do not have access to the FTP application.
10. Create an alias so that mail sent to the FTP user is sent to an administrator account.
11. Use of the firewall precluded configuration of tcpwrappers on the server to allow only specified addresses to have access.
12. Install Tripwire® after the server was configured. The use of this advanced file versioning system adds to confidence that the server has not been compromised.
13. Require SSH access to perform maintenance. The administrators that maintain the server are required to use this secure shell access because the servers exist in an untrusted network.

Firewall Configuration:

As with any implementation of firewalls, the goal is delineate and protect segments of the network while providing authorized access to services. In this implementation, it is necessary to either provide a single firewall with three network interfaces (logical separation), or two firewalls with dual network interfaces (physical and logical) to provide the isolated DMZ. The FTP server must go in the DMZ that is created by the firewall(s). Placing the FTP server in the DMZ allows isolation of the FTP server from both the Internet and the trusted intranet. This provides protection for the FTP server from the Internet and protects the trusted intranet from the FTP server. To complete the isolation of the FTP server, the firewall configurations only allow the FTP server to be a destination for network traffic from either the Internet or intranet, it may never serve as a source.

There is no desire to provide an FTP server for any use other than our business file transfers. The firewall provides access controls based on source, service and destination. This allows F100C to limit the number of sites that can connect to the FTP server. It allows limiting the traffic to the FTP server to only the FTP protocol. This enforcement does require that all participants have static Internet addresses. Since the firewall allows only traffic to the FTP server, it provides a mitigating control over the FTP server if it were compromised, confining it to the DMZ.

Overview of a file transfer:
The steps in the encryption process start and finish with the unencrypted file on a server or mainframe computer located inside a trusted network. It is important to note, that the data stored on the FTP server in the DMZ is always a duplicate of the file stored (and backed up) within a trusted zone. As an example, starting with a file located in the F100C's trusted network destined for the vendor site the following points are steps in the file transfer process:

1. First the file is encrypted with the public key for the target vendor on a server, mainframe or workstation in the trusted environment.
2. When the encryption is completed, file is transferred through the firewall to the inbound directory on the FTP server in the DMZ. The users inbound directory of the FTP server in the DMZ is write only for the FTP user.
3. At regular intervals, the status of any files found in this directories are checked, and if the file is closed, the first and last lines of the file are checked to see that they conform to the standard header and trailer records for PGP encrypted files. If the file meets these criteria, it is then moved to an outbound directory that is accessible by the vendor. If it does not meet these criteria, it is discarded and the event is logged.
4. At regular intervals, the vendor FTP initiates a connection through vendor's firewall and then through the local firewall to the DMZ FTP server. The connection is accepted or rejected at the local firewall based on source IP address.
5. The vendor then logs on to the FTP server. If there is a file available, it is uploaded to vendor FTP server.
6. The vendor is to remove the file from the directory after a successful upload. If a file stays on the FTP server beyond agreed to limits, it is deleted and the event is logged.
7. The file is decrypted at the vendor's trusted site and is now available for their use.

FTP server's local file handling cron job script features:
The actual script used is written in perl. Below, is a shell script that demonstrates the file handling and some of the logging that is done (echo statements) to move the files from inbound directories to outbound and provide the cursory check that the files are PGP® encrypted.

```
#!/bin/sh
FTPFILENAMES=/tmp/results
LOCALHOMEIN=/tmp/test
LOCALHOMEOUT=/tmp/outbound
HEADERRECORD="BEGIN PGP MESSAGE"
TRAILERRECORD="END PGP MESSAGE"
 # find all the files that have not been accessed for at least one minute
# storing the names in a file is not necessary, but easier to explain
echo $(find $LOCALHOMEIN -type f -a -amin +1) > $FTPFILENAMES
 # this section reviews the list, inspects each file and enforces encryption
# echo statements would be good places to log the activity of this script
if [[ -r $FTPFILENAMES ]]
then
  exec 4< $FTPFILENAMES
  while read -u4 line
  do
    for f in $line
    do
      echo "Found a file name $f"
      if head -1 $f | grep -q -e "$HEADERRECORD" &&
        tail -1 $f | grep -q -e "$TRAILERRECORD";
      then
        if [[ !$(/bin/mv $f $LOCALHOMEOUT) ]]
        then
          echo "Successfully moved $f"
        else
          echo "Move of $f failed."
        fi
      else
        echo "Not PGP encrypted - deleting $f"
        rm -f $f
      fi
    done
  done
  exec 4<&-
else
  echo "$FTPFILENAMES  is not readable"
fi
```

Sample PGP® encrypted file below shows the expected format for any file:

```
-----BEGIN PGP MESSAGE-----
Version: PGP 7.0.4

qANQR1DDDQQDAwJ85oWxFykAI2DJO0mhNlM0XB3QaZSj+944Sk2USeBlwi1S4KE2
EuPywR9F43aZYvPoyxMrfbkkKPJz+NB9Tnrpohvg+a8e
=VpYG
-----END PGP MESSAGE-----
```

This process shown above has three distinct components. First verify that the file is not actively being accessed by any other process. Next do a simple verification that the file fits the encrypted model for PGP. Finally either move the file if valid or delete it if it does not have the proper format. To date, a scan of the server has reliably been performed at five minute intervals to enforce the business rules described in the script.

In an effort to follow the published practices for hardening a FTP server[2], all users are only allowed to put files into write-only directories on the server, and only allowed to read and delete from the outbound directory. As can be seen in the script above, the files are inspected to verify that they have proper PGP® header and trailer records, moving or deleting the file as appropriate. This simple security check verifies that the proper encryption is being performed. The logging is important to help isolate the reason for the files that may not get delivered because of a system problem that caused a file to get through without encryption. The PGP® check is also valuable to keep control of the server and keep out business partners that are not willing to PGP® encrypt the data files. This check has been extremely valuable to the support areas.

Missing from the script are the components that change permissions based on the destination of the file, logic that is used to determine the appropriate outbound directory, and the cleanup script that removes old data files.

The file names have in them the key to the destination for the file. This must be negotiated between the business partners and the technicians that are responsible for creating the files

Host and network IDS

A method was needed to monitor the integrity of the FTP server. A Network Intrusion Detection System (NIDS) tap was placed in the DMZ to monitor the activity of the FTP server. The rule is pretty simple for the NIDS, the FTP should not be the source for any communication. Also, Tripwire® was installed on the FTP server to notify F100C of any unexpected changes in the file system of the server. A backup of the FTP servers are performed anytime that there is a maintenance update of the server.

High availability:

High availability for the FTP process was a requirement that surfaced late in the implementation. The need was driven by business partners that saw this as a critical part of the business process and by support areas that needed to be able to provide maintenance of the FTP servers.

Efforts to load balance the FTP servers have failed. Currently F100C has pairs of servers implemented in active and hot standby at the primary site. At the secondary site, both servers are ready and would be brought up if the primary site were to become unusable.

Business recovery:
The definition of high availability was finally modified to provide for disaster recovery. As part of the implementation, F100C also attempted to implement global load balancing across two geographically separate locations in addition to local load balancing. If one data center was disabled, another one 900 miles away would probably still be functional. The original goal was to not only provide the site as a "hot standby", but rather load balance across the Internet. At the onset, F100C had two vendors with proposed solutions that gave a very high confidence level that such a task was simple. After six months of failures, F100C gave up on the ability to provide anything more than hot site capability. Both vendors have revamped their solutions and promise to now be able to provide this capability. Even though the capability would be nice, there is no compelling business driver to revisit or purse the Global Load Balancing at this time.

Summary of key points from this implementation:

1. One year after implementation of the Internet FTP process, F100C has about a dozen heavy users. They see additional vendors being added at about two per month.
2. The bulk of the specialized processing is bourn by F100C.
3. The tightest timing interval for updates has proven to be about five minutes.
4. The IDS system sends an alarm if any traffic has a source of the FTP server in the DMZ.
5. Firewalls block the FTP server if it attempts to originate a connection.
6. Vendor to Vendor use of the FTP server in the DMZ is blocked, unless specifically authorized by the F100C.
7. The files transferred through this DMZ FTP server must be PGP encrypted.
8. The address of the vendor client FTP server must be known by F100C to allow them access. As a note, this is the most fragile part of the entire process. People who know the FTP process at vendor sites seem to rarely know what address they are actually presenting to the Internet.

9. The second most difficult part of this process is the exchange and changing of PGP keys. The public PGP keys used are not put onto public PGP key servers, they are held in confidence by the companies involved in the exchange of files.
10. The actual environment consists of at least 2 servers capable of performing each task to provide high availability
11. Finally, the entire environment is located at two geographically separate sites to provide for disaster recovery.

Future plans:
At this time everyone at F100C is satisfied with the performance of the current process and especially with its reliability. The vendors who use this method are very agreeable with the current model used as it does not force them to perform and special tasks other than to encrypt/decrypt the files with PGP®. The process of adding a new vendor into the system is getting smoother.

There is some interest in using a Virtual Private Network (VPN) connection to eliminate the need for the static Internet address, however the IP Security Protocol (IPSec) tunneling is not mature and lacks a standard for interoperability between vendor implementations. Without the standard, I do not see a change to use the VPN for transfers. The process of choosing vendors for VPN tunneling software and hardware would set off another round of negotiations over the licensing of the VPN software, liability associated with connecting between the environments, and with additional support for the VPN.

SSH/SCP is also being considered, but it also brings in the new vendor product issues and all the legal and support negotiations with it. There is enough interest in the use of SSH/SCP to expect that it will be adopted some time in the future.

No matter what process is used to enhance the security of the connection, the encryption of the data as a separate process will remain as an additional layer to mitigate the risk of sending the data over the Internet.

SANS, “Internet DMZ equipment policy” URL:
http://www.sans.org/resources/policies/Internet_DMZ_Equipment_Policy.pdf

Fortune, “Fortune 100 companies”, June 21, 2002, URL:
http://www.hrc.org/equalityatexxon/fortune.asp

“The File Transfer Protocol (FTP) service” URL:
http://www-uxsup.csx.cam.ac.uk/security/probing/about/anonftp.html

Garfinkel, Simson & Spafford, Gene. Practical Unix & Internet Security. Sebastopol:O’Reilly & Associates, Inc., April 1996. 487 – 494.

Latis Networks. “The Data Protection Rule of the GLBA: A strategy for compliance.” 2003 URL:
http://www.stillsecure.com/index.jsp?sector=papers_briefs

Hewlett-Packard Company. “downloads, drivers and notifications”. 2003 URL:
http://welcome.hp.com/country/us/eng/software_drivers.htm

International Business Machines. “All downloads and drivers”. URL:
http://www-1.ibm.com/support/all_download_drivers.html

Internet Engineering Task Force. “IP Security Protocol (ipsec)”. 10 April 2003 URL:
http://www.ietf.org/html.charters/ipsec-charter.html

HIPAA.ORG. “HIPAA Final Security Rule”. 20 Feb 2003. URL:
http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=2003_register&docid=fr20fe03-4

RSA Security. “What is Diffie-Hellman?”. 2003. URL:
http://www.rsasecurity.com/rsalabs/faq/3-6-1.html

NIDS:
http://www.robertgraham.com/pubs/network-intrusion-detection.html#1.1

Office of Thrift Supervision. “Laws and Regulations”. 30 May 2003. URL:
http://www.ots.treas.gov/pagehtml.cfm?catNumber=17

Price, Will. “Inside PGP® Key Reconstruction”. May 2003. URL:
http://www.pgp.com/products/whitepapers/PGPKeyRecon.pdf

Benz, Jessica. “PGP A Hybrid Solution”. 2001. URL:
http://www.sans.org/rr/papers/20/717.pdf

RSA Security. “RSA-based Cryptographic Schemes”. 2003. URL:
http://www.rsasecurity.com/rsalabs/rsa_algorithm/index.html

SANS. “SANS/FBI Top Twenty List Version 3.22”. 3 March 2003. URL:
http://www.sans.org/top20/#U5

Schneider, Neil. “TCPWRAPPERS”. 30 May 2003. URL:
http://www.linuxgeek.net/index.pl/tcpwrappers

Tripwire. “Tripwire for servers Functional Details”. 2003. URL:
http://www.tripwire.com/products/servers/functional.cfm?