



SANS Institute

Information Security Reading Room

Case Study In Secure File Transfer: Implementing Secure FTP with SSL In a Healthcare Organization

Steve Tobias

Copyright SANS Institute 2021. Author Retains Full Rights.

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

**Case Study In Secure File Transfer:
Implementing Secure FTP with SSL
In a Healthcare Organization**

GIAC Security Essentials Certification (GSEC)
Practical Assignment Version 1.4b - Option 2

Steve Tobias
July 14, 2004

© SANS Institute 2004, Author retains full rights.

Table of contents

1.0	Abstract	1
2.0	Before Snapshot – Problem Description	1
2.1	General	1
2.2	Before-Methods Of Access	2
2.3	Before-Configuration	2
2.4	Before-Security Posture	4
3.0	During – Analysis	7
3.1	SSH	7
3.2	SSL	8
3.3	FTP Protocol Basics	10
3.4	FTP and SSL	10
4.0	Product Choice	11
5.0	During – The Implementation	11
5.1	Design Architecture	11
5.2	Hardening the FTP Servers	12
5.2.1	Rename Administrator account	13
5.2.2	Apply Latest System OS Patches, Install Antivirus Software	13
5.2.3	Windows Security Templates	13
5.2.4	Steps to Apply Windows Security Template	13
5.2.6	Verify Services and Ports	17
6.0	Tumbleweed Installation and Configuration	21
6.1	Install Front-end Streaming Proxy	21
6.2	Install Back-end Data Management Server	23
6.3	Tumbleweed Configuration	23
6.3.1	TCP/IP Ports Used	23
6.3.1	Authentication Configuration	24
6.3.2	File Transfer Options	24
6.4	Secure FTP Client Testing	25
6.5	Firewall Configuration	27
	Conclusion	27
	References	29
	Appendix A	32
	Appendix B	33
	Appendix C	34
	Additional Reading	36

List of Figures

Figure 1 – Before State FTP Configuration.....	3
Figure 2 – SSL/TLS OSI Model Relationship	9
Figure 3 – Tumbleweed Secure Transport Architecture	12
Figure 4 – Security Templates	14
Figure 5 – Activate Template Configuration	15
Figure 6 – Template Install Status.....	15
Figure 7 – Secure Transport Installation	22
Figure 8 – Secure Transport Port Configuration	22
Figure 9 – Root Certificate Generation.....	23
Figure 10 – Secure Transport TCP/IP Ports.....	24
Figure 11 – Secure Transport Client.....	25
Figure 12 – WS_FTP Pro Client	25
Figure 13 – MoveIT Freely Command Line Client.....	26
Figure 14 – Ethereal trace showing Explicit FTPS connection sequence	26

List of Tables

Before State Risk Assessment	4
Project Requirements.....	5
TCP/IP Stack Hardening.....	17
AFD.sys Denial of Service Hardening	17
Sample Listing Secure FTP Server Products	32
Sample Listing Secure FTP Clients	32

© SANS Institute, Author retains full rights.

1.0 Abstract

Secure electronic file transfer between organizations has become essential for business transactions and communication. Healthcare organizations are no exception to this requirement. The ability to leverage the Internet to share protected health information also known as PHI or other sensitive information between healthcare organizations is ever increasing. From individual file encryption and VPN's (Virtual Private Networks), to a complete EDI (Electronic Data Interchange) system, a plethora of methods and applications exist for securing the transfer of files and data over the Internet.

This case study presents the implementation of secure file transfer using FTP over SSL (File Transfer Protocol over Secure Sockets Layer)¹ in a healthcare organization – a project for which I was technical lead. Before state, project requirements including risk assessment, reasoning behind product selection, implementation, and technical information regarding FTP, SSL and SSH (Secure Shell) will be presented. Satisfying HIPAA (Health Insurance Portability and Accountability Act)² requirements will also be touched on. Research and consideration were given to several different methods for secure file transfer including a complete EDI solution. Due to the specific project requirements of the healthcare organization, the solution chosen was a highly customizable and scalable product that uses FTP over SSL with the additional ability of file encryption. Finally, an evaluation of the chosen solution including mitigation of risk factors will be discussed.

2.0 Before Snapshot – Problem Description

2.1 General

Selected as technical lead for a project to replace an existing file transfer system for a large health care organization, it was my job to analyze, recommend and implement an upgrade solution. The existing file transfer system was a somewhat kludged together system that was comprised of a Windows 2000 server running a third party standard FTP server application, a scheduler utility, and a file/directory monitoring utility. The FTP server essentially functioned as a transparent gateway for data interchange between core back-end systems and provided limited access from the Internet from health care organization partners. The FTP server was a member server in a mixed Novell Netware/NT Domain environment where the NT Domain was interconnected to Novell Netware via a special redirector installed on the NT domain controllers.

¹ FTP, SSL – Also see Sections 3.2, 3.3

² HIPAA –Also see Appendix B Definition of Terms

2.2 Before-Methods Of Access

Access to the FTP server was permitted not only via standard FTP from internal trusted systems, but also via network shares configured on the FTP server itself.

Limited FTP access from the Internet was also devised. FTP access was permitted by any standard FTP client application on the trusted LAN with a valid FTP user name and password.

Network share level access was allowed to the FTP data directories on the server. Workstations had the ability to map drives to certain shares on the server that applications running on the users workstations could utilize. Permissions to shares were assigned using NT Domain Global groups. Users were placed into Global groups and Global groups given access rights.

Internet access to the server was eventually deemed an urgency – and so two methods were utilized to allow limited secure access to the server from the Internet:

- a) Clientless VPN (browser-based; Microsoft Internet Explorer required)
- b) Standard VPN client (for both Windows and Unix platforms)

Browser-based VPN access was used for user or manual initiated file transfers (one or just a few files), while standard VPN client access was used when automated or scripted file transfers (many files, scheduled basis) were required on the partner side.

2.3 Before-Configuration

The third party FTP application used only it's own user database and had no provision to interface into any Directory Services such as LDAP³ (Light Weight Directory Access Protocol) or an NT Domain forcing yet another desperate user database to administer and monitor.

The FTP server itself maintained a number of mapped drives and share level access to other internal Windows and Netware servers (the Novell Netware Client was installed on the server to allow connections to Netware servers). Any mapped drive or network path the operating system could connect to – the FTP server application could also connect to. For several processes this allowed the FTP server to function as a pass-through file transfer gateway. An example of this would be a Unix system needing to FTP a file to a Netware system. Since the FTP server had the appropriate drive mapped to the Netware destination (usually a common file share directory) the FTP user for the Unix system would FTP files to a virtual FTP directory pointing to the mapped Netware drive. The virtual directory simply appeared to be a standard directory from an FTP perspective.

The FTP server application was configured to run in a Domain Administrator account context to enable it to connect to any other server shares required for FTP processes. Because of limitations with running the FTP application as a service interactively,

³ LDAP – Appendix B Definition of Terms

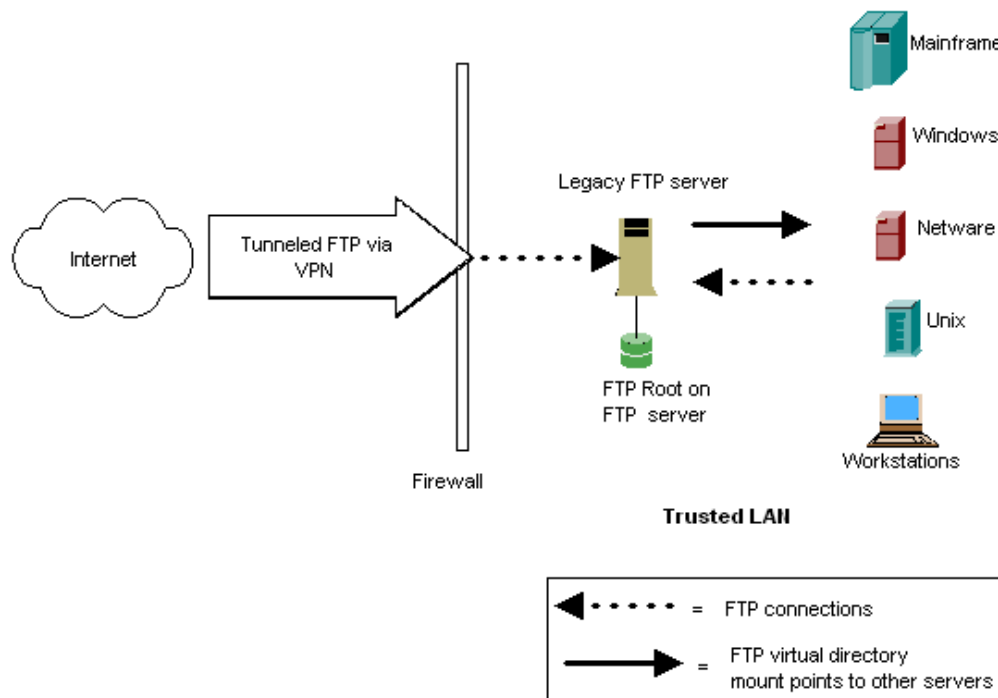
automatic logon to the FTP server using a Domain Administrator account was required to allow the FTP server application to interact with the desktop.

A separate scheduler utility was used to move files on a regular basis by way of FTP and standard file copy. This was also used for self-monitoring of connections to other servers and other maintenance tasks. A third utility for directory and file monitoring was used in combination with homegrown scripts for FTP processes that required additional operations such as zip, unzip, or file renaming or GnuPG encryption.

Internet access for browser-based connections and standard VPN client connections were complex to configure and fell under the responsibility of respectively two separate teams.

Located on the trusted LAN and originally implemented to only handle several internal system-to-system file transfers, requests for internal file transfers as well as partner access from the Internet quickly exceeded the capacity, function and design of the server. The server hardware being six years old and out of warranty, also begged to be replaced. The diagram below shows the before state FTP server, its connections to other internal servers and systems as well as typical file transfer flow.

Figure 1 – Before State FTP Configuration



2.4 Before-Security Posture

Defining before state risks and potential vulnerabilities were done based partly on consensus based information covered in the SANS Security Essentials Course and from the National Institute of Standards and Technology Computer Resource Center publication: Risk Management Guide for Information Technology Systems ⁴.

In identifying, controlling and minimizing any negative impact associated with each risk, primary risks were identified and associated with one of three categories:

1. Confidentiality
2. Integrity
3. Availability

A risk assessment table below lists the major risks defined by category, negative impact rating based on probability of occurrence and severity of consequences and risk options (to accept, reduce or transfer the risk).

Table 1
Before State Risk Assessment

Identified Risk	Category	Impact Rating	Risk Options
Old Hardware	Availability	High	Reduce
Auto logon – Domain Admin account	Integrity, Confidentiality, Availability	High	Reduce
Single Server	Availability	Medium	Reduce
Complexity of Remote Partner Access	Availability, Confidentiality	High	Reduce
Non integration of user database to other systems	Confidentiality, Integrity	Medium	Reduce
Separate utilities for file & maintenance functions	Integrity, Availability	Low	Accept / Reduce if possible or Transfer
HIPAA ⁵ Compliance	Integrity, Confidentiality, Availability	High	Reduce

⁴ <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>

⁵ HIPAA – See Appendix B Definition of Terms

Impact ratings of medium and high were top priority to investigate solutions for. As noted in the Risk Management Guide for Information Technology Systems:

For a system that is in the initiation or design phase, system information can be derived from the design or requirements document. For an IT system under development, it is necessary to define key security rules and attributes planned for the future IT system. System design documents and the system security plan can provide useful information about the security of an IT system that is in development⁶.

Therefore, combining a strong knowledge of the organization's business operations with the additional goal of enhancing the security posture of the file transfer system, the actual project requirements were developed and refined.

Table 2 lists actual project requirements and matches the original risk where possible to denote how project requirements were aligned to minimize or address certain risks. Note that some project requirements were business operations based and therefore not necessarily matched directly to a specific risk.

Table 2
Project Requirements

Project Requirement	Matching Risk Corrected
New Server Hardware	Old Hardware
Application required to be run as a "service"	Auto logon – Domain Admin account
Used server "virtualization" tools to create separate development and production systems	Single server
Secure Connections / File Transfer	Complexity of Remote Partner Access HIPAA Compliance
Choice to encrypt/not encrypt data residing on FTP server (data at rest)	Complexity of Remote Partner Access HIPAA Compliance

⁶ <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>, 11

Project Requirement	Matching Risk Corrected
Batch processing/automation: Event driven, Schedule driven	Separate utilities for file & maintenance functions
LDAP/SSO/Directory Services integration	Non integration of user database to other systems HIPAA Compliance
RFC ⁷ 2228 FTP security extensions compliance. http://www.ietf.org/rfc/rfc2228.txt	HIPAA Compliance
Auto report generation – web based reports, trending, usage, and failures	Business Need
Administrative Delegation	Business Need
Web based: administration, customizable/ branding – portal look and feel	Business Need
User space quotas/restrictions	Business Need
Hide files/directories – by user ID, Group	HIPAA Compliance
Proprietary client not required	Business Need
Standard FTP support for internal / Back-end processes	Business Need

⁷ RFC – Request For Comments

Project Requirement	Matching Risk Corrected
FTP/S and HTTP/S file transfers	HIPAA Compliance, Business Need
Run on Win32 platform (Windows 2000 / 2003)	Business Need

3.0 During – Analysis

Significant research was performed regarding enterprise level file transfer methods that would satisfy project requirements. The solution required would need to be implemented in a matter of months, minimize impact on back-end system-to-system transfers by being compatible with standard FTP as well as meet other project requirements for secure internet based connectivity and especially HIPAA Compliance.

A complete EDI (Electronic Data Interchange) system was originally envisioned, however, cost and time constraints quickly scaled back the research to only target methods of secure FTP or similar file copy methods.

Investigation regarding secure file transfer over the Internet revealed SSH (Secure Shell, also known as Secure Socket Shell) and SSL (Secure Socket Layer) as the two current primary options used for secure file transfer communications. As of this writing equally viable products exist for enhanced secure file transfer using either SSH or SSL. (See Appendix A for sample listing of secure FTP products.)

Both SSH and SSL can accomplish session traffic encryption and connection authentication using industry standard encryption algorithms such as RSA key exchange and Triple DES.

3.1 SSH

SSH was originally designed as a replacement for unsecured applications such as telnet, rlogin, rsh and ftp where usernames and passwords are sent in clear text across a network. It can also be used to securely “tunnel” other applications. The standard TCP/IP port used for SSH is 22. SSH and its associated components are applications that can perform a variety of tasks.

More information regarding SSH can be found at the following links:

<http://www.openssh.com/>
<http://www.ietf.org/html.charters/secsh-charter.html>
<http://www.ssh.com/support/cryptography/>
<http://www.onsight.com/faq/ssh/ssh-faq-1.html#ss1.1>

3.2 SSL

SSL was originally designed by Netscape Corporation, as an Internet browser add-on (as opposed to an “application” in the case of SSH) for secure web communications. SSL is a universally accepted standard for secure web based transactions such as credit card purchases and other ecommerce. It typically uses TCP/IP port 443.

TLS (Transport Layer Security) protocol, based on SSL, has since superceded SSL as the official standard while retaining backward compatibility with SSL versions 2.0 and 3.0. Many times the protocols are referred to together as SSL/TLS.

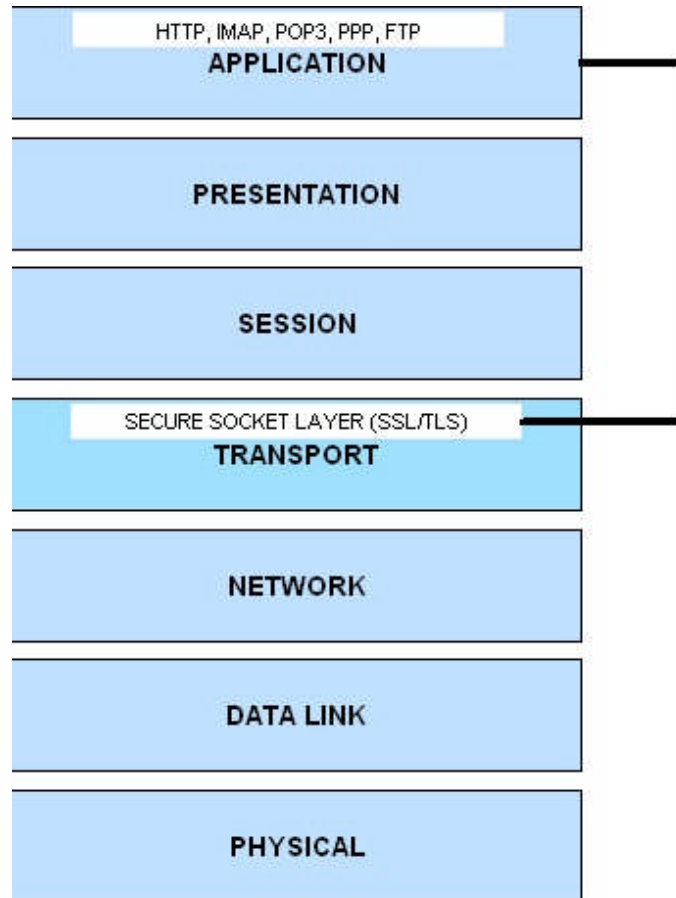
RFC 2246 is the base Internet standards document for TLS and states:

The primary goal of the TLS Protocol is to provide privacy and data integrity between two communicating applications...The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.⁸

SSL itself is not an application. Operating at the Transport layer of the OSI (Open Systems Interconnection) model it provides services to other higher layer application protocols, functioning as an application independent method for confidential, authenticated, integrity based communication between applications. Figure 2 depicts SSL/TLS in logical relation to other applications using the seven layer OSI model.

⁸ <http://www.ietf.org/rfc/rfc2246.txt>

Figure 2 – SSL/TLS OSI Model Relationship



Applications must be designed to use SSL/TLS for which there are several SSL related standards, which are beyond the scope of this study. Additional information regarding SSL/TLS can be found at the following links:

- <http://www.mozilla.org/projects/security/pki/nss/ssl/>
- <http://developer.netscape.com/tech/security/ssl/howitworks.html>
- <http://wp.netscape.com/security/techbriefs/ssl.html>
- <http://www.ietf.org/html.charters/tls-charter.html>
- <http://www.kegel.com/ssl/>
- <http://www.openssl.org/>
- <http://www.stunnel.org/>

3.3 FTP Protocol Basics

The FTP protocol uses two TCP/IP (Transmission Control Protocol/Internet Protocol) connections or channels. The control channel uses port 21 for the initial connection; the data channel uses port 20 for the data transfer connection.

Two modes of operation can be used:

- Active
- Passive

In Active mode the FTP client sets a random high port (>1024) for the data channel that the FTP server should initiate a connection to from its port 20. In Passive mode the FTP client initiates both the control and data channel connections to the FTP server. The FTP client opens two high ports (N>1024, N+1). Port N connects to port 21 on the FTP server, sends a PASV command to the server to inform it that it is in passive mode, the server then tells the FTP client what high port (P>1024) it would like to use to setup the data channel connection. The FTP client then opens a connection from its N+1 port to the FTP server's high port P for data transfer.⁹ A major security vulnerability exists with standard FTP in that usernames and passwords are sent in clear text. FTP is listed in the SANS Top 20 Internet Security Vulnerabilities¹⁰.

Note: Passive mode is considered to be more Firewall friendly; as the Passive FTP port range can usually be configured within the FTP server application.

3.4 FTP and SSL

Securing FTP using SSL can be referred to as FTP over SSL, FTP/S or FTPS. (Securing FTP with SSH is typically referred to as Secure FTP or SFTP.)

Typically one of two possible modes is used for FTP over SSL:

- Explicit SSL/TLS – AUTH SSL, AUTH TLS: connection starts on standard FTP port 21, switches to SSL or TLS based on FTP client requesting SSL encryption via AUTH SSL or AUTH TLS command respectively. Standards compliant to RFC 2228 - FTP Security Extensions.
- Implicit SSL/TLS – FTP connection starts on a designated port (usually 990), SSL is started at the beginning of the connection. As of this writing the IETF (Internet Engineering Task Force) has not formally adopted an RFC for Implicit SSL/TLS. Potential standards are covered in several Internet drafts. Explicit SSL should be used where standards compliance is mandated.

⁹ <http://slacksite.com/other/ftp.html>

¹⁰ <http://www.sans.org/top20/>

Some excellent resources for further information on FTP/SSL are listed below.

http://www.cuteftp.com/support/WebHelp/Explicit_versus_implicit_SS.htm

<http://www.indyproject.org/KB/index.html?howdoiuseftpwithssl.htm>

<http://www.mozilla.org/projects/security/pki/nss/ssl/>

<http://wp.netscape.com/security/techbriefs/ssl.html>

<http://www.ietf.org/html.charters/tls-charter.html>

<http://www.ietf.org/rfc/rfc2246.txt>

<http://www.ietf.org/rfc/rfc2228.txt>

4.0 Product Choice

Based on project and security requirements the vendor/application choice was narrowed to two possibilities, both of which had excellent features and strengths:

Tumbleweed's Secure Transport¹¹ and Standard Network's MoveIT DMZ product¹². Both products utilized SSL/TLS, however MoveIT DMZ could also use SSH. Both products were RFC 2228¹³ standards compliant for FTP security extensions. While MoveIT DMZ was less than half the cost of Secure Transport and was similar in function and capability, Tumbleweed's Secure Transport application architecture and administration features more closely matched the project requirements for the organization. Secure Transport was also more flexible and scalable as it pertained to the organization's business needs. Tumbleweed's Secure Transport Enterprise edition was chosen for implementation.

5.0 During – The Implementation

5.1 Design Architecture

Tumbleweed's Secure Transport Enterprise Edition is a dual server configuration. A Front-end FTPS / HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) proxy server that streams data to a Back-end data management server. Only authentication and connections occur on the Front-end proxy, data and files never placed on this server; but are streamed to the Back-end data management server. The Front-end proxy server component resides in the DMZ (Demilitarized Zone) while the Back-end data management server resides in the trusted LAN. Standard FTP is allowed only to the Back-end server. The FTP root directory was configured to exist on a clustered Netware volume that has relative access to corporate common shared data areas. This greatly enhanced the file transfer and user access manageability on the trusted LAN pertaining to file rights and permissions. Virtual FTP directory mount points to other healthcare organization servers are also enabled. Internet based users

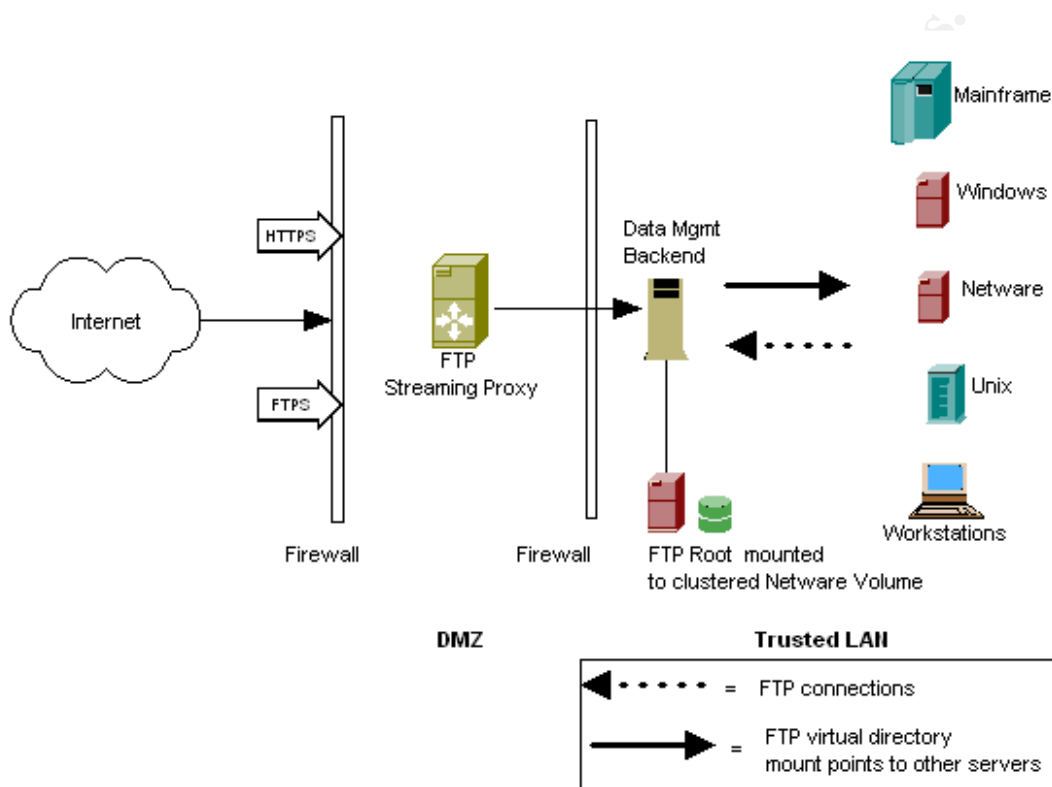
¹¹ http://www.tumbleweed.com/products/securetransport_form.html

¹² http://www.stdnet.com/products/?category_number=2&subcategory_number=1

¹³ <http://www.ietf.org/rfc/rfc2228.txt>

have the choice of using any RFC 2228 compliant FTPS or HTTPS client for file transfer. Figure 3 depicts the new file transfer architecture.

Figure 3 – Tumbleweed Secure Transport Architecture



5.2 Hardening the FTP Servers

Microsoft Windows Server 2003 was chosen for both the Front-end proxy and Back-end data management server components. Compared to previous versions of Windows released by Microsoft, Windows Server 2003 is more secure by default, however, the attack surface can still further be reduced by following best practices to improve the security state and in some cases the operational state of a default Windows 2003 install especially for the server located in the DMZ exposed to the internet.

Summary of steps used to harden Windows Server 2003

1. Rename Administrator account
2. Apply latest system OS patches
3. Install Antivirus software
4. Apply modified security template (twFTP- Server Baseline.inf)
5. Verify successful application of security template

6. Disable NetBIOS¹⁴ over TCP/IP (FTP Front-end proxy only)
7. Scan server with vulnerability tools
8. Install Tumbleweed FTP software
9. Re-scan server with vulnerability tools

5.2.1 Rename Administrator account

Renaming the default Administrator account on the server, although it may be considered security-by-obscurity, was done inline with best practices. A decoy Administrator account was also created as recommended in Hacking Exposed Windows Server 2003¹⁵.

5.2.2 Apply Latest System OS Patches, Install Antivirus Software

Latest Microsoft patches and Antivirus software were installed. These two steps are mandatory in our organization on any server build. Also considered best practice.

5.2.3 Windows Security Templates

The Windows 2003 Sever Security Guide¹⁶ and it's companion guide, Threats and Countermeasures¹⁷, provide a wealth of information, tools, templates, test scripts, best practices and security related recommendations. In preparation to utilize enhanced registry settings made available in the security templates provided with these guides, the Security Configuration Editor User Interface was updated according to instructions on pages 241–245 of the Threats and Countermeasures Guide. The High Security - Member Server Baseline.inf template that came with the Windows 2003 Server Security Guide was used with some slight modifications to fit the health care organization's requirements. The beginning portion of the actual template .inf file is listed in Appendix C to illustrate the format used in the .inf file. Hardening Windows Systems¹⁸ also gave great insight into the use of Windows Security Guides and Templates.

5.2.4 Steps to Apply Windows Security Template

Important: Applying security templates can be very difficult to reverse. Make sure you have a good backup, image or undo template created before applying templates. Secedit /GenerateRollback¹⁹ can be used to create an undo template.

The Security Configuration Editor MMC snap-in was used to apply the template however secedit from the Windows 2003 Resource Kit could have been used as well.

1. Copy Security Template .inf file to %systemroot%\security\templates.

¹⁴ Network Basic Input Output System – Also see Appendix B Definition of Terms

¹⁵ Scambray/McClure, 132

¹⁶ <http://www.microsoft.com/downloads/details.aspx?FamilyID=8a2643c1-0685-4d89-b65521ea6c7b4db&displaylang>

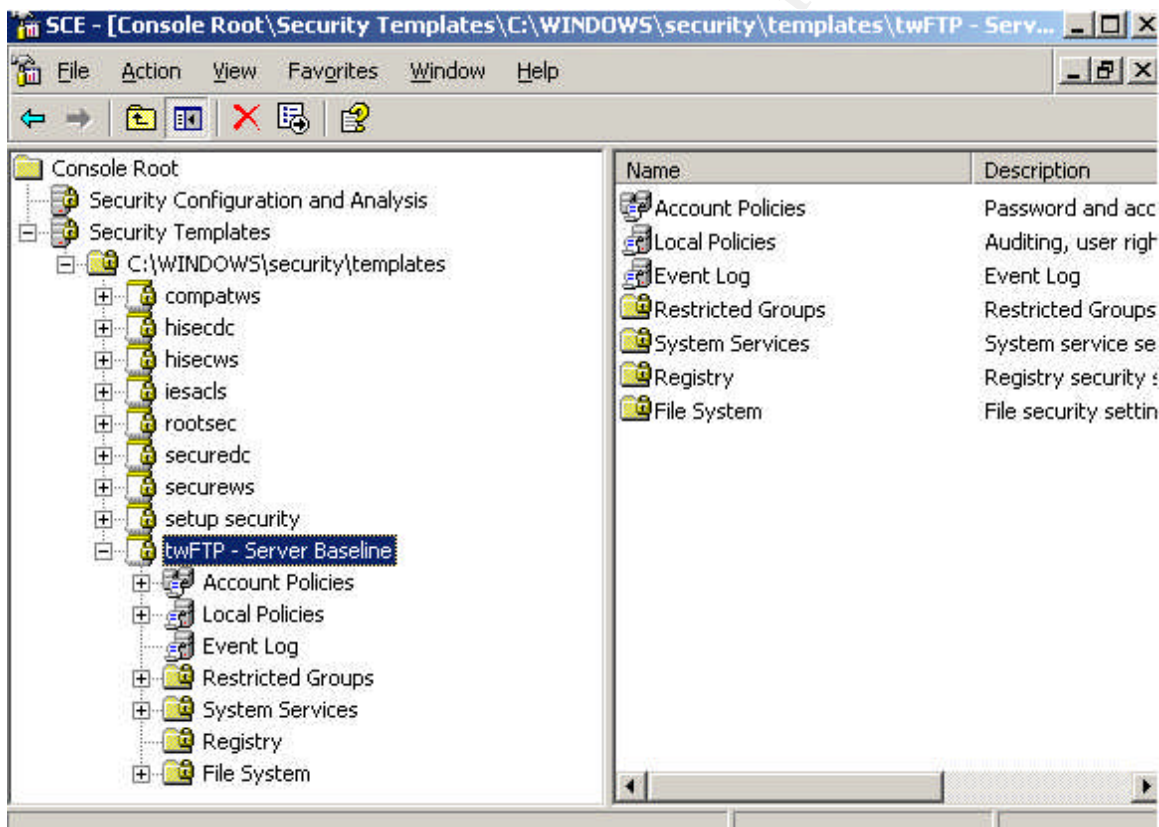
¹⁷ <http://www.microsoft.com/technet/security/topics/hardsys/tcg/tcgch00.mspx>

¹⁸ Bragg, 222-293

¹⁹ Bragg, 268-269

2. Update Security Configuration Editor .inf and re-register .dll per Threats and Countermeasures Guide²⁰.
3. Load Security Templates MMC snap-in:
 - 3.0 Start | Run | MMC | File | Add/Remove Snap-in.
 - 3.0.1. Add | Security Templates, and Security Configuration and Analysis.
Tip: Save as SCE.msc for future use. Run from Administrative Tools.
4. Expand Security Templates left pane.
The newly copied security template should show up in the list as shown in the screen shot below.

Figure 4 – Security Templates

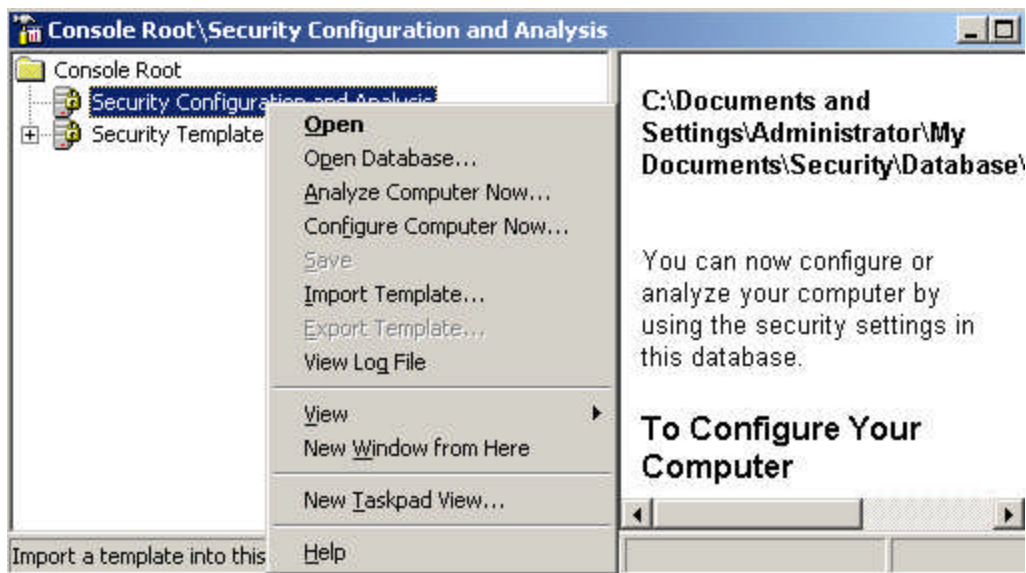


5. Right click Security Configuration and Analysis | Select Open Database.
6. Type a name for the new database | Click Open.
7. Import Template window will appear | Select new .inf template | Click Open.

²⁰ <http://www.microsoft.com/technet/security/topics/hardsys/tcg/tcgch00.mspx> , 241-245

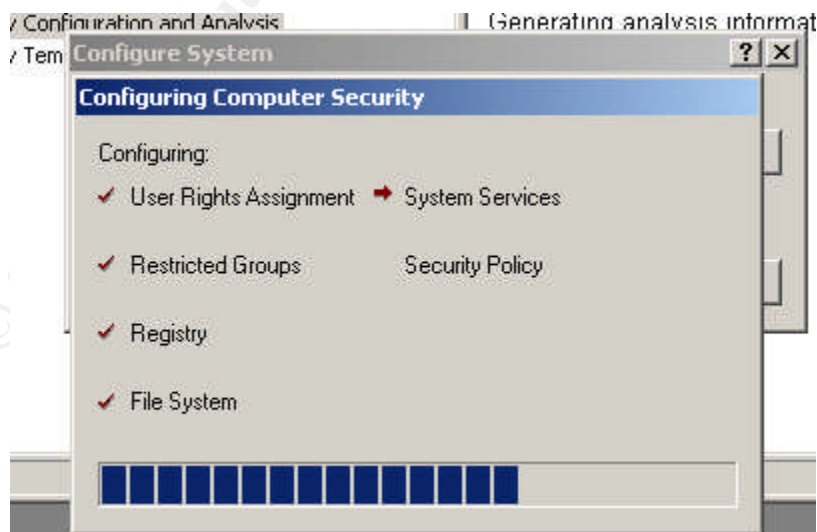
8. Right click Security Configuration and Analysis | Select Configure Computer Now.

Figure 5 – Activate Template Configuration



9. Accept and note the default log location.
10. Status screen will display during template application followed by a log report in the right window pane once the security template has been applied.

Figure 6 – Template Install Status



11. Verify log from step 9. for template application errors. It should look similar to the abbreviated version shown below.

Log file output showing Security Template application results:

Log file: C:\Documents and Settings\Administrator\My Documents\Security\Logs\twFTPServers.log

Tuesday, April 13, 2004 10:53:28 AM

----Configuration engine was initialized successfully.----

----Reading Configuration Template info...

----Configure User Rights...

Configure S-1-5-20.

Configure S-1-5-19.

Configure S-1-5-32-544.

Configure S-1-5-32-546.

Configure S-1-5-7.

Configure S-1-5-32-547.

Configure S-1-5-32-551.

Configure S-1-5-11.

User Rights configuration was completed successfully.

.
.

Configure machine\system\currentcontrolset\services\tcpip\parameters\tcpmaxdataretransmissions.

Configure machine\system\currentcontrolset\services\tcpip\parameters\tcpmaxportsexhausted.

Configure machine\system\software\microsoft\windows nt\currentversion\winlogon\screensavergraceperiod.

Configuration of Registry Values was completed successfully.

----Configure available attachment engines...

Configuration of attachment engines was completed successfully.

----Un-initialize configuration engine...|



5.2.5 Security Template Settings

Detailed explanation of all of the settings in this template are beyond the scope of this study, however, a summary noting important settings accomplished by using the security template as well as Tables detailing TCP/IP stack hardening are listed below.

Windows 2003 Security Template Settings Summary:

- TCP/IP stack hardening against Denial of Service / SYN²¹ attacks
- NetBIOS over TCP/IP turned off (performed manually – not via template)
- Logon banner – legal warnings
- Don't display last user logon
- Unused system services minimized / disabled based on server role
- Event log 90% full warning
- Disable 8.3 file name auto generation

Table 3
TCP/IP Stack Hardening

Registry Key	Value
EnableICMPRedirect	0
SynAttackProtect DWORD	1
EnableDeadGWDetect DWORD	0
EnablePMTUDiscovery	0
KeepAliveTime	300000
DisableIPSourceRouting	2
TcpMaxConnectResponseRetransmissions	2
TcpMaxDataRetransmissions	3
PerformRouterDiscovery	0
TCPMaxPortsExhausted	5

Table 4
AFD.sys Denial of Service Hardening

Registry Key	Value
DynamicBacklogGrowthDelta	10
EnableDynamicBacklog	1
MinimumDynamicBacklog	20
MaximumDynamicBacklog	20000

The Windows 2003 Sever Security Guide and Threats and Countermeasures Guide have detailed information on the use and modification of Windows Security Templates.

5.2.6 Verify Services and Ports

System services and ports were verified before and after applying the modified security template. Netstat, Net Start and Systemals pslist²² were used locally on the server.

²¹ SYN – “Synchronize” ; part of the TCP/IP connection establishment sequence. A SYN attack can be used to tie up TCP/IP resources causing a denial of service on a system.

²² <http://www.sysinternals.com/ntw2k/freeware/pstools.shtml>

Nmap²³ and Gfi's LANgaurd Network Security Scanner²⁴ were used to scan for vulnerabilities from the network. Scan results shown are from the Tumbleweed Front-end proxy server with the Tumbleweed FTP/SSL services running, post security template application.

Running Net Start from a command line displays currently running Windows services. Output of Net Start command showing services running.

```
C:\Documents and Settings\Administrator>net start
These Windows services are started:
```

```
Automatic Updates
COM+ Event System
Computer Browser
Cryptographic Services
Cygwin cron
DHCP Client
DNS Client
Event Log
IPSEC Services
Network Connections
Network Location Awareness (NLA)
NT LM Security Support Provider
Plug and Play
Protected Storage
Remote Procedure Call (RPC)
Remote Registry
Security Accounts Manager
Server
System Event Notification
TCP/IP NetBIOS Helper
Terminal Services
Tumbleweed_admind
Tumbleweed_agentd
Tumbleweed_ftpd
Tumbleweed_httpd
VMware Tools Service
Windows Management Instrumentation
Windows Time
Workstation
```

```
The command completed successfully
```

²³ <http://insecure.org/nmap>

²⁴ <http://www.gfi.com/lannetscan/>

Netstat displays TCP/IP connections. “-a” shows all listening ports and connections, “-n” shows addresses/ports as numbers, “-o” (new to Windows Server 2003 and XP) shows the owning process ID (PID). Output of Netstat -ano command showing listening ports mapped to process ID's:

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING	1444
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING	1492
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	696
TCP	0.0.0.0:443	0.0.0.0:0	LISTENING	1492
TCP	0.0.0.0:444	0.0.0.0:0	LISTENING	1532
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:1029	0.0.0.0:0	LISTENING	524
TCP	0.0.0.0:4455	0.0.0.0:0	LISTENING	1340
TCP	127.0.0.1:1025	0.0.0.0:0	LISTENING	1340
TCP	127.0.0.1:1027	0.0.0.0:0	LISTENING	1444
TCP	127.0.0.1:1385	0.0.0.0:0	LISTENING	3612
UDP	0.0.0.0:445	*:*		4
UDP	0.0.0.0:500	*:*		524
UDP	0.0.0.0:1026	*:*		880
UDP	0.0.0.0:1028	*:*		952
UDP	0.0.0.0:4500	*:*		524
UDP	10.10.42.20:123	*:*		952
UDP	127.0.0.1:123	*:*		952
UDP	192.168.116.129:123	*:*		952

Pstlist is like an enhanced version of net start with several options. Output of the pstlist command showing process names and ID's:

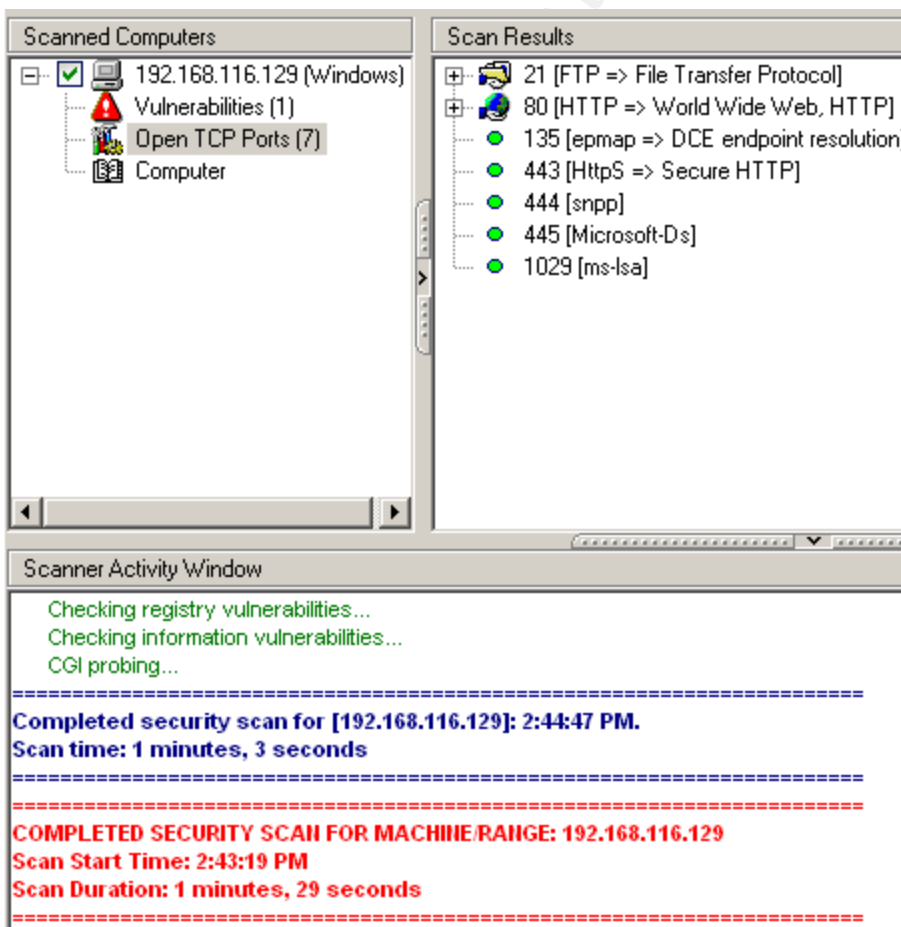
Name	Pid	Pri	Thd	Hnd	Priv	CPU Time	Elapsed Time
Idle	0	0	1	0	0	0:49:56.562	0:00:00.000
System	4	8	47	1194	0	0:00:30.515	0:00:00.000
smss	288	11	3	17	164	0:00:00.531	0:53:10.828
csrss	444	13	12	520	1748	0:00:26.546	0:53:05.031
winlogon	468	13	17	450	5744	0:00:05.125	0:52:44.359
services	512	9	16	279	1368	0:00:03.390	0:52:43.593
lsass	524	9	21	369	7228	0:00:06.843	0:52:43.421
svchost	696	8	10	157	844	0:00:01.015	0:52:42.703
svchost	752	8	16	128	1300	0:00:05.984	0:52:42.281
svchost	880	8	7	119	3228	0:00:00.250	0:52:39.765
svchost	908	8	5	76	572	0:00:00.078	0:52:39.640
svchost	952	8	26	608	8888	0:00:04.421	0:52:39.546
cygrunsvr	1004	8	4	62	1120	0:00:00.171	0:52:38.843
cron	1128	8	2	52	1064	0:00:00.296	0:52:38.078
svchost	1180	8	2	33	268	0:00:00.031	0:52:37.546
cygrunsvr	1204	8	4	62	1120	0:00:00.312	0:52:37.468
cygrunsvr	1236	8	4	62	1120	0:00:00.343	0:52:37.390
cygrunsvr	1272	8	4	62	1120	0:00:00.218	0:52:37.296
cygsrv	1280	8	3	56	1080	0:00:00.218	0:52:37.265
cygrunsvr	1320	8	4	62	1120	0:00:00.343	0:52:37.156
sh	1332	8	3	53	1068	0:00:00.453	0:52:37.125
agentd	1340	8	3	79	2572	0:00:00.484	0:52:37.093
VMwareService	1380	13	3	33	396	0:00:10.781	0:52:36.921
sh	1396	8	4	55	1092	0:00:00.296	0:52:36.875
ftpd	1444	8	3	82	2716	0:00:00.640	0:52:36.656
httpd	1492	8	3	121	3272	0:00:02.375	0:52:35.890
admin	1532	8	1	83	6376	0:00:01.921	0:52:34.687
httpd	1676	8	4	103	3548	0:00:00.484	0:52:30.234
httpd	1692	8	4	103	3548	0:00:00.421	0:52:30.109
httpd	1708	8	4	103	3548	0:00:00.375	0:52:29.906
httpd	1732	8	4	103	3548	0:00:00.421	0:52:29.687
httpd	1756	8	4	103	3548	0:00:00.421	0:52:29.296
admin	316	8	101	186	6112	0:00:02.609	0:52:23.703

Nmap has many scanning options. “-sS” performs a TCP:SYN scan; the first part of a TCP connection is attempted and then closed if a reply received. “-O” attempts to guess or “fingerprint” the target operating system.

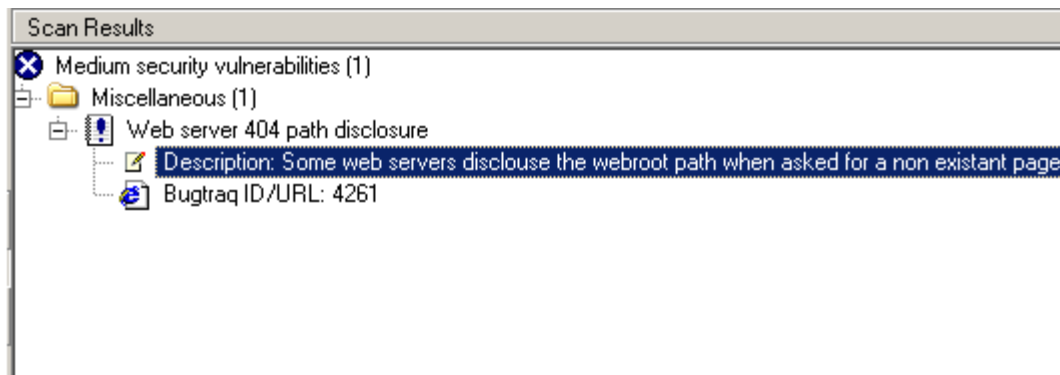
Output of Nmap -sS -O showing open ports:

```
Starting nmap V. 3.10ALPHA4 ( www.insecure.org/nmap/ )
Interesting ports on 192.168.116.129:
(The 1598 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open      ftp
80/tcp    open      http
135/tcp   open      loc-srv
443/tcp   open      https
444/tcp   open      snpp
445/tcp   open      microsoft-ds
1029/tcp  open      ms-lsa
No exact OS matches for host (If you know what OS is running on it, see http://www.insecure.org)
TCP/IP fingerprint:
SInfo(V=3.10ALPHA4P=i586-pc-linux-gnuD=7/13Time=40F3A870D=21C=1)
TSeq(Class=TR%IPID=I%TS=0)
T1(Resp=Y%DF=N%M=4000%ACK=S++%Flags=AS%Ops=MNNNT)
T2(Resp=Y%DF=N%M=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=N%M=4000%ACK=S++%Flags=AS%Ops=MNNNT)
T4(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLen=B0%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Output of LANguard 5.0 showing open ports:



Output of LANguard showing potential CGI vulnerability:



The CGI path disclosure vulnerability could not be reproduced however Tumbleweed has been notified of the above findings. This was considered an acceptable risk assuming the vendor will remedy the path disclosure issue if one is truly found.

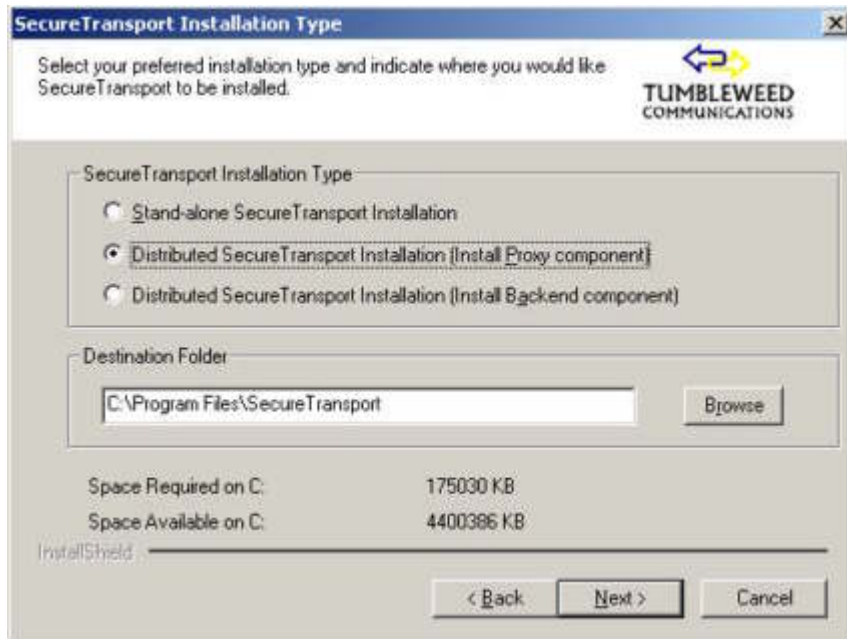
6.0 Tumbleweed Installation and Configuration

Tumbleweed's Secure Transport Enterprise Edition is a dual server configuration. A Front-end FTP/S, HTTP/S proxy server that streams data to the Back-end data management server. It is advisable to install and configure the Back-end server first due to configuration questions that will need to be answered when installing the Front-end. Decisions also need to be made whether or not to use self signed Certificates generated during the install routine or use Certificates from a trusted third party authority. Certificate exchange must occur between Front-end and Back-end servers before streaming can occur. The healthcare organization chose to use a third party authority Certificate on the Front-end only; as this is the Internet facing side where customer contact occurs. Please see the Additional Reading section for resource links on Certificates. Additionally TCP/IP port numbers will be required for various components that comprise communications between servers for data streaming, SSL, FTP, HTTP, HTTPS and Administration. Since both server installs are more or less mirrors of each other only the Front-end proxy server installation steps are shown. Secure Transport is shipped with several Installation and Administration Guides, which provide detailed information for installation and configuration.

6.1 Install Front-end Streaming Proxy

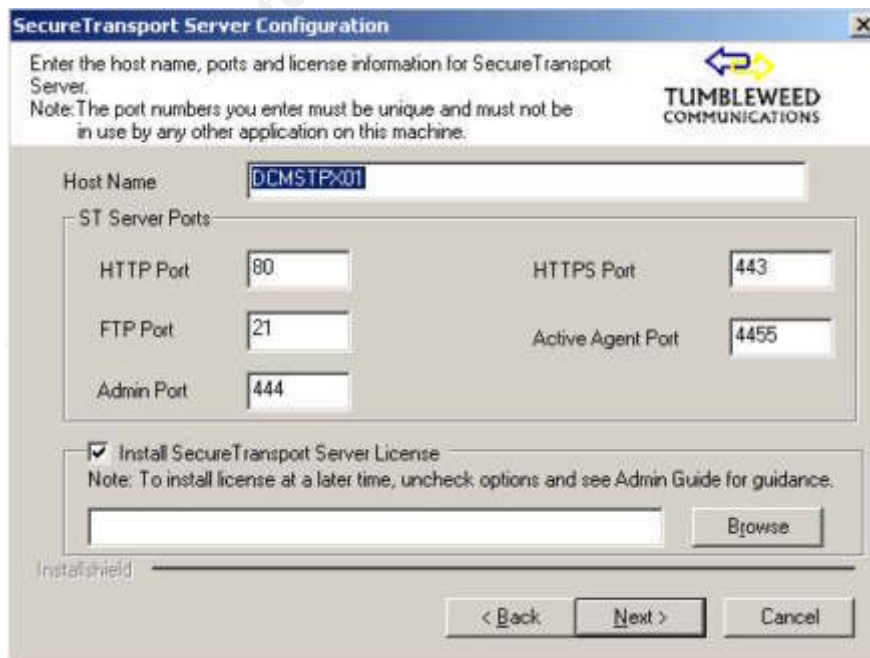
1. Run Secure Transport setup.exe
2. Select installation type: Proxy or Back-end component:

Figure 7 – Secure Transport Installation



3. Accept or modify ports as required:

Figure 8 – Secure Transport Port Configuration



4. Root Certificate generation is required even if you plan to import a third party Certificate later.

Figure 9 – Root Certificate Generation

Root Certificate Settings

SecureTransport requires a root certificate for generating all Server Certificates used by SecureTransport. Enter the information below for generating the root certificate.

TUMBLEWEED COMMUNICATIONS

Root Certificate Settings

Private Key Password: [masked]

Verify Password: [masked]

Validity Period: 3650 days

Private Key Length: 1024 bits

Installshield

< Back Next > Cancel

5. After installation is complete use the Tumbleweed supplied gencsr.pl script and steps outlined in the Installation Guide to generate the Certificate Request File: csr-req.pem in order to request and then install third party CA Certificates - replacing the Secure Transport generated Certificates.

6.2 Install Back-end Data Management Server

In addition to above steps, the Secure Transport Back-end data management server also has installation and configuration screens for a Transaction Manager component and file encryption options.

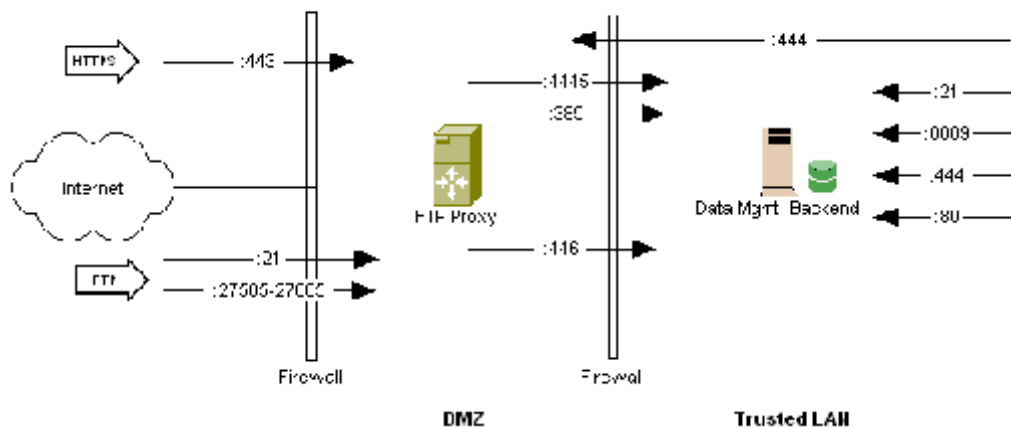
6.3 Tumbleweed Configuration

6.3.1 TCP/IP Ports Used

Secure Transport uses the following TCP/IP ports as configured for the healthcare organization. (Also see Figure 10.)

- 21 FTP (initial connection only)
- 389 LDAP connection to trusted LAN
- 443 HTTPS File Transfer
- 444 HTTPS Web Based Administration
- 4445 Active Agent Port (data streaming)
- 27505 – 27805 Passive FTP/SSL Port Range
- 80 HTTP File Transfer – Back-end only

Figure 10 – Secure Transport TCP/IP Ports



6.3.1 Authentication Configuration

Authentication options include Virtual Users that are unknown to the operating system created within Secure Transport, LDAP (including Active Directory) Users, and System Users from the operating system (Unix or Windows). To accomplish the upgrade of the old FTP system the healthcare organization chose to initially use Virtual Users, as this was the scheme of the existing FTP system. All users will eventually be migrated to LDAP users.

6.3.2 File Transfer Options

File transfer options include FTPS and HTTPS from Internet clients to the Front-end proxy, and FTP or HTTP from trusted LAN clients to the Back-end. Several FTPS clients were successfully tested with the Front-end proxy. In testing, it was discovered that some clients preferred Passive Mode FTP, which is also generally more Firewall compatible. Passive Mode FTP was eventually enabled on the FTP Front-end proxy and the port range set to 27505 – 27805.

6.4 Secure FTP Client Testing

Screen shots of some of the FTP clients tested are shown in a connected state below. Also shown is an Ethereal trace showing the typical FTP/SSL connection sequence going from port 21 to SSL with an AUTH SSL command. Appendix A contains a sample listing of FTP/SSL Clients and Servers.

Figure 11 – Secure Transport Client



Figure 12 – WS_FTP Pro Client

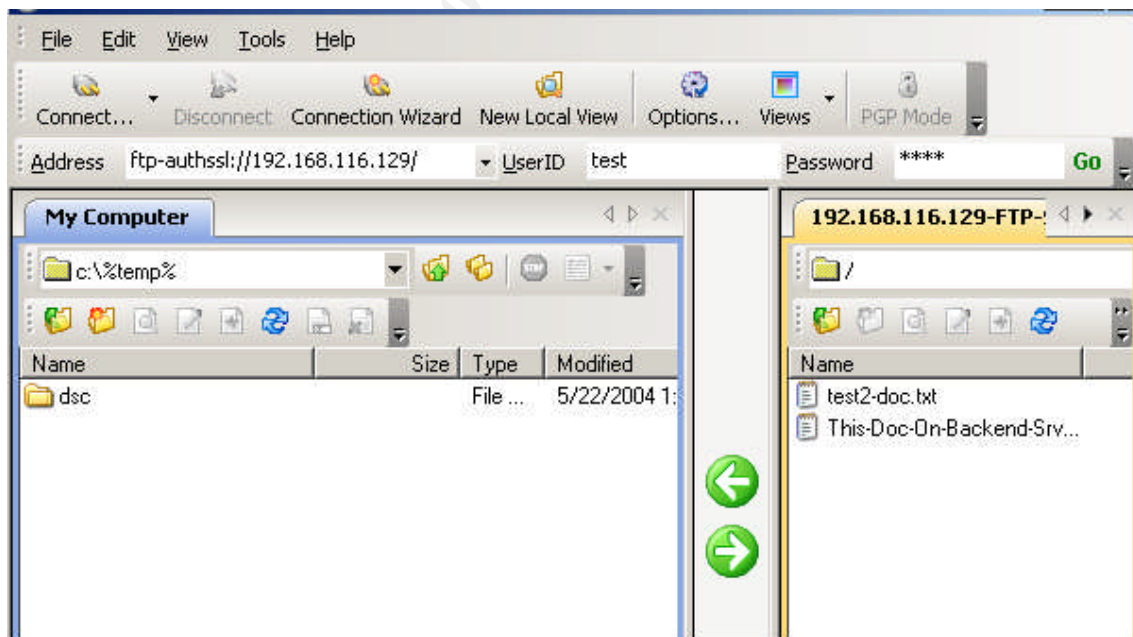


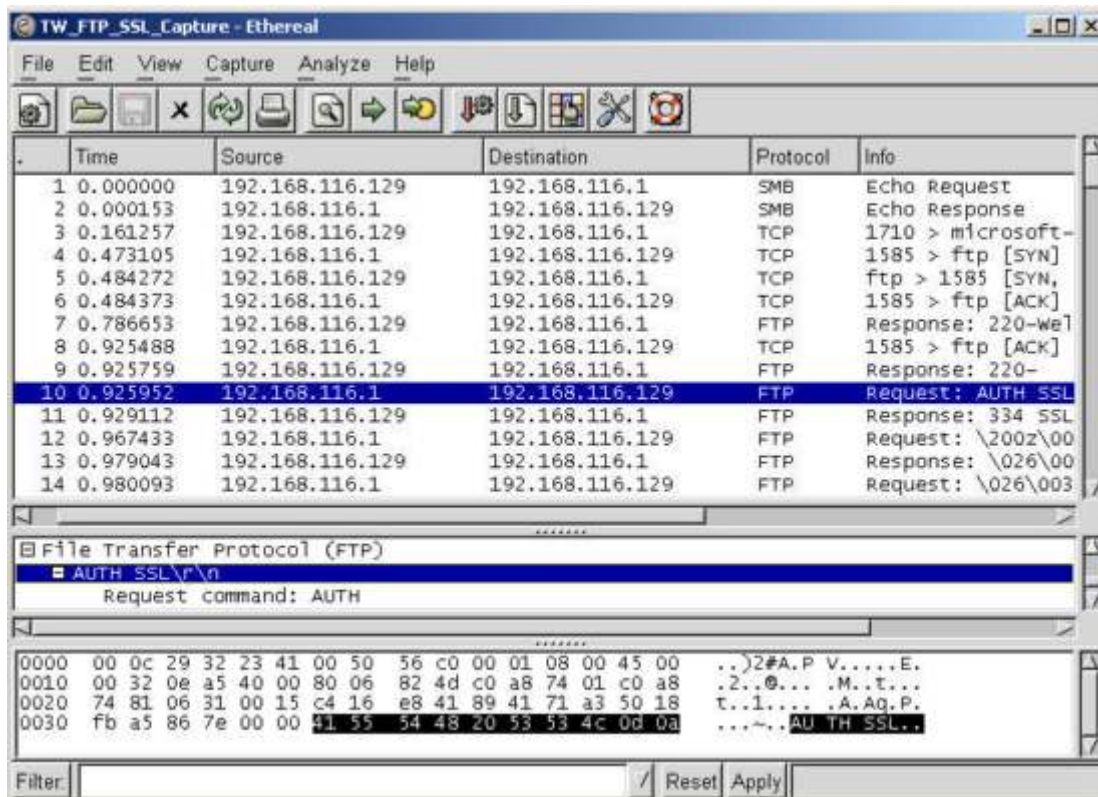
Figure 13 – MoveIT Freely²⁵ Command Line Client

```

C:\>ftps -d -e:on 192.168.116.129
220-Welcome to Health Care Org Secure File Transfer
220-
220 Secure FTP Server ready.
---> AUTH TLS-P
334 SSLv23/TLSv1
Connected to 192.168.116.129.
User: test
---> USER test
331 Password required for test.
Password: ****
---> PASS ****
230 Virtual user test logged in.
---> SYST--->
215 Cygwin Type: LB
ftp> _

```

Figure 14 – Ethereal²⁶ trace showing Explicit FTPS connection sequence



²⁵ <http://demos.stdnet.com/moveitfreely/>

²⁶ <http://www.ethereal.com>

6.5 Firewall Configuration

The healthcare organization's Firewall was configured as follows:

FTP Front-end proxy (DMZ)

- Inbound < Internet: TCP: 21, 443, 27505 – 27805
- Inbound < Trusted LAN: TCP: 444
- Outbound > Back-end Server: TCP: 446, 4445
- Outbound > LDAP Server: TCP: 389

An external DNS name was also assigned to one of the organizations registered IP addresses for the secure FTP site.

7.0 High Availability – Redundancy

High availability in the form of a cold standby is currently being achieved by using server virtualization software from VMware called VMware ESX Server²⁷. Both the Front-end proxy and Back-end data management servers are virtualized server nodes running on ESX server. A snap-shot (as well as normal system backups) is periodically taken of the server nodes, which are essentially each just one big file. Either server can be restored to another unused ESX session or a physically separate ESX server in the case of hardware failure in a matter of minutes.

Conclusion

The implementation of the Tumbleweed file transfer system occurred with only a few complications. Licensing of Secure Transport is done by embedding the IP address of the server to which the product will be installed. The healthcare organization changed the IP address assigned the Front-end server to a different one than originally licensed. The product will not function with out a valid license that matches the IP embedded in the license itself. It was discovered and later noted in the product documentation that at least one of the IP addresses assigned to the server was required to match the license file. Once the original IP was added to the server, transfers occurred via the additional IP address (not listed in the license file) without incidence. Although not required, a replacement license file could also be obtained from the vendor. The complex task of migrating the healthcare organizations vendors, partners and back-end systems to the new Tumbleweed system continues as of this writing. The newly added ability in Windows Server 2003 to create an undo security template is a definite plus. This was not easily accomplished with previous versions of Windows. No complications between Secure Transport and the Windows security template were observed.

²⁷ http://www.vmware.com/products/server/esx_features.html

The Table below displays risk mitigation details showing how the individually identified risks were addressed during the project.

Identified Risk	Category	Impact Rating	How Mitigated
Old Hardware	Availability	High	New Hardware Purchased
Auto logon – Domain Admin account	Integrity, Confidentiality, Availability	High	Secure Transport modules run as “services”
Single Server	Availability	Medium	Dual server configuration, server virtualization with VMware
Complexity of Remote Partner Access	Availability, Confidentiality	High	Industry standard FTPS clients or standard internet browser (HTTPS) can be used
Non integration of user database to other systems	Confidentiality, Integrity	Medium	LDAP interface built into Secure Transport
Separate utilities for file & maintenance functions	Integrity, Availability	Low	Secure Transport Transaction Manager module; rule based data management / task engine
HIPAA Compliance	Integrity, Confidentiality, Availability	High	“Reasonably Secure “ SSL connections, extensive logging, digitally signed audit trails / MDN receipts, LDAP interface, optional data (at rest) encryption.

Overall security risk has been reduced or addressed. Recommendations to reduce risks even further for the Front-end proxy server include but are not limited to:
Complete removal of the SMB (Server Message Block) protocol, enabling EFS (Encrypted File System), using IPSec (Internet Protocol Security) filters as a host-based firewall or installing a third party host-based firewall.

The choice of Tumbleweed Secure Transport Enterprise Edition to enable robust and scalable secure file transfer for the healthcare organization was a definite success. Implementation of the product greatly enhanced the security posture and manageability of the healthcare organization's file transfer system. The Front-end data streaming feature is notable in that it adds an additional layer of defense from the Internet. Regulatory compliance for HIPAA standards is currently being achieved. Project requirements and business needs have also been reasonably met. Tumbleweed's additional features such as MDN (Message Disposition Notification) receipts and file encryption may be used by the healthcare organization in the future pending further corporate security policy review and establishment.

References

Public Law 104-191 “Health Insurance Portability And Accountability Act of 1996”
21 August 1996 URL: <http://aspe.hhs.gov/admnsimp/pl104191.htm> (21 May 2004)

Department of Health and Human Services, Office of the Secretary, 45 CFR Part 162:
“Administrative Simplification: Standard Unique Health Identifier for Health Care
Providers; Final Rule”. 23 January 2004. URL:
<http://a257.g.akamaitech.net/7/257/2422/14mar20010800/edocket.access.gpo.gov/2004/pdf/04-1149.pdf> (21 May 2004)

Jupitermedia Corporation. “Online Dictionary” URL:
<http://www.webopedia.com/> (7 July 2004)

Stoneburner, Gary. Goguen, Alice. Feringa, Alexis. National Institute of Standards and
Technology Computer Resource Center “Risk Management Guide for Information
Technology Systems” July 2002. URL:
<http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf> (25 June 2004)

Various Authors. “Secure Shell (secsh) Charter”. 3 November 2003. URL:
<http://www.ietf.org/html.charters/secsh-charter.html> (22 May 2004)

SSH Communications Security. “Cryptography A-Z”. URL:
<http://www.ssh.com/support/cryptography/> (8 July 2004)

Acheson, Steve. “The Secure Shell Frequently Asked Questions”. URL:
<http://www.onsight.com/faq/ssh/ssh-faq-1.html#ss1.1> (7 July 2004)

Deirks, T. Allen, C. “Request for Comments: 2246 The TLS Protocol Version 1.0”
January 1999. URL: <http://www.ietf.org/rfc/rfc2246.txt> (22 June 2004)

“Active FTP vs. Passive FTP, a Definitive Explanation”. URL:
<http://slacksite.com/other/ftp.html> (22 June 2004)

Standard Networks, “MoveIT Freely FTP Client Documentation”. URL:
<http://demos.stdnet.com/moveitfreely/> (22 June 2004)

The Mozilla Organization, “SSL/TLS” 3 May 2002. URL:
<http://www.mozilla.org/projects/security/pki/nss/ssl/> (7 May 2004)

Netscape Communications Corporation. “How SSL Works” URL:
<http://developer.netscape.com/tech/security/ssl/howitworks.html>
(10 June 2004)

Netscape Communications Corporation. "Secure Sockets Layer" URL: <http://wp.netscape.com/security/techbriefs/ssl.html> (10 June 2004)

Netscape Communications Corporation. "Introduction to SSL" URL: <http://developer.netscape.com/docs/manuals/security/sslin/contents.htm> (10 June 2004)

Reynolds, J. Postel, J. "Request for Comments: 959 File Transfer Protocol (FTP)" October 1985. URL: <http://www.ietf.org/rfc/rfc959.txt> (10 May 2004)

Internet Engineering Task Force. "Transport Layer Security (tls)" 14 November 2003. URL: <http://www.ietf.org/html.charters/tls-charter.html> (10 June 2004)

Bellovin, S. "Request for Comments: 1579 Firewall-Friendly FTP". February 1994. URL: <http://www.rfc-archive.org/getrfc.php?rfc=1579> (10 June 2004)

Kegel, Dan. "SSL / TLS" 25 September 2001. URL: <http://www.kegel.com/ssl/> (10 June 2004)

"Open SSL Frequently Asked Questions". URL: <http://www.openssl.org/support/faq.html> (10 June 2004)

"Stunnel Frequently Asked Questions". URL: <http://www.stunnel.org/faq/> (25 June 2005)

GlobalSCAPE. "Explicit versus Implicit SSL". URL: http://www.cuteftp.com/support/WebHelp/Explicit_versus_implicit_SS.htm (25 June 2004)

Hower, Chad. "What is the difference between implicit TLS and explicit TLS?". URL: <http://www.indyproject.org/KB/index.html?whatisthedifferencebetweentls.htm> (7 July 2004)

Hower, Chad. "How do I use FTP with SSL?". URL: <http://www.indyproject.org/KB/index.html?howdoiuseftpwithssl.htm> (7 July 2004)

Horowitz, M. Lunt, S. "Request for Comments: 2228 FTP Security Extensions" October 1997. <http://www.ietf.org/rfc/rfc2228.txt> (7 July 2004)

Tumbleweed Communications. "Tumbleweed Secure Transport". URL: http://www.tumbleweed.com/products/securetransport_form.html (10 June 2004)

Standard Networks. "MoveIT DMZ Overview". URL: http://www.stdnet.com/products/?category_number=2&subcategory_number=1 (10 June 2004)

"Nmap Network Security Scanner Man Page". URL: http://www.insecure.org/nmap/data/nmap_manpage.html (26 June 2004)

“GFI LANguard Network Security Scanner”. URL:
<http://www.gfi.com/lannetscan/> (26 June 2004)

“Ethereal 0.10.5 distribution”. 7 July 2004. URL:
<http://www.ethereal.com/distribution/> (10 July 2004)

Sharp, Richard. “Ethereal User's Guide”. URL:
<http://www.ethereal.com/docs/user-guide/> (23 June 2004)

Internet Numbers Assigned Authority. “Port Numbers”. 15 May 2004. URL:
<http://www.iana.org/assignments/port-numbers> (21 May 2004)

Ipswitch, Inc. “SSL Encryption”. URL:
http://www.ipswitch.com/Products/WS_FTP/B2B/ssl.html (10 June 2004)

Patrick, Rob. “What is the difference between SSH and SSL?”. 28 October 2003. URL:
<http://www.rpatrick.com/tech/ssh-ssl/> (10 June 2004)

Columbia University. “The Kermit FTP Client - Secure Scriptable FTP”.
October 2002. <http://www.columbia.edu/kermit/ftpclient.html> (1 June 2004)

Vmware/EMC. “VMware ESX Server 2.1”. URL:
http://www.vmware.com/products/server/esx_features.html (24 June 2004)

Microsoft Corporation. “Windows Server 2003 Security Guide”. 28 January 2004. URL:
<http://www.microsoft.com/downloads/details.aspx?FamilyID=8a2643c1-0685-4d89-b655-521ea6c7b4db&displaylang> (21 May 2004)

Microsoft Corporation. “Threats and Countermeasures Guide” 23 April 2003. URL:
<http://www.microsoft.com/technet/security/topics/hardsys/tcg/tcgch00.mspx>
(21 May 2004)

SANS Institute, “The SANS Top 20 Internet Security Vulnerabilities” Version 4.0.
8 October 2003. <http://www.sans.org/top20/> (12 June 2004)

Scambray, Joel / McClure Stuart. Hacking Exposed Windows Server 2003.
Emeryville: McGraw-Hill/Osborne, 2003.

Bragg, Roberta. Hardening Windows Systems. Emeryville:
McGraw-Hill/Osborne, 2004

Appendix A

Sample Listing Secure FTP Server Products

FTP Server	Link	Encryption Used
Tumbleweed: Secure Transport	http://www.tumbleweed.com/products/securetransport_form.html	SSL (FTP/S, HTTP/S)
Standard Networks: MoveIT DMZ	http://www.stdnet.com/products/?category_number=1&subcategory_number=1	SSL, SSH (FTP/S, SFTP, HTTP/S)
Vandyke Software: vShell Server	http://www.vandyke.com/products/vshell/index.html	SSH (SFTP)
F-secure SSH	http://f-secure.com/products/fssh/	SSH (SFTP)
Ipswitch: WS_FTP Server	http://www.ipswitch.com/products/ws_ftp_server/index.html	SSL (FTP/S)
GlubTech: Secure FTP Wrapper	http://www.glub.com/	SSL (FTP/S)
Berkeley University of California: SafeTP (wrapper)	http://safetp.cs.berkeley.edu/	SSL (FTP/S)

Sample Listing Secure FTP Clients

FTP Client	Link	Encryption Used
Tumbleweed: Secure Transport	http://www.tumbleweed.com/products/securetransport/clients.html	SSL (FTP/S, HTTP/S)
Ipswitch: WS_FTP Pro	http://www.ipswitch.com/Products/WS_FTP/features.html	SSL / SSH / PGP(files) (FTP/S, SFTP, +)
SourceForge: FileZilla	http://filezilla.sourceforge.net/	SSL / SSH (FTP/S, SFTP)
GlobalSCAPE: CuteFTP Pro	http://www.cuteftp.com/cuteftp/compare.asp http://www.cuteftp.com/cuteftppro/	SSL / SSH / OTP(S-key) (FTP/S, SFTP, +)
CORE FTP	http://www.coreftp.com/	SSL / SSH (FTP/S, SFTP)
Columbia University: C-Kermit	http://www.columbia.edu/kermit/ftpclient.html	SSL / Kerberos 4,5 / GSSAPI / SRP (FTP/S, +)
Curl	http://curl.haxx.se/	SSL + (FTP/S, HTTP/S)
Standard Networks: MoveIT Freely	http://demos.stdnet.com/moveitfreely/	SSL

Appendix B

Definition of Terms

HIPAA - Health Insurance Portability and Accountability Act, signed into law 1996. Several revisions over the last few years have occurred. Currently the law sets technology independent standards at the federal level for safeguarding electronic protected health information, ePHI or just PHI - for healthcare organizations and health care providers. Separate portions of these standards are being phased in over a period of several years and as of this writing will culminate in May of 2007. The law imposes civil and or financial penalties for non-compliance.

EDI – Electronic Data Interchange, the exchange of data between corporations using a network.

LDAP – Light Weight Directory Access Protocol, set of open standards protocols for accessing information directories.

NetBIOS - Network Basic Input Output System, an API that augments the DOS BIOS by adding special functions for local-area networks (LANs).²⁸ It can be configured to run on top of other protocols such as TCP/IP.

Certificates

There are many resources on the Internet regarding Digital Certificates and Public Key Infrastructure (PKI) and the potential complexity for implantation thereof. Some of these are noted in “Additional Reading” at the end of this paper. Digital Certificates are an electronic means to prove or verify identities. A digital pair of keys (a public key and a private key) are assigned to an identify (person, corporation etc.) that can be used to digitally sign and/or encrypt data.

Certificates and SSL/TLS:

Columbia University²⁹ had a very simplified and succinct explanation of Certificates and SSL/TLS.

When SSL/TLS is used to provide security, authentication of the server and optionally the client can be performed using X.509 Public Key Certificates. Certificates are used to exchange a public key for use in establishing an encrypted connection and can be verified against a known trusted Root Certificate and a Certificate Revocation List (CRL) to indicate its authenticity and validity. The contents of the certificate can then be used to determine the identity of the remote service or the client.

²⁸ <http://www.webopedia.com/>

²⁹ <http://www.columbia.edu/kermit/security81.html#xa3.1>

Appendix C

Security Template File Used to Harden Windows Server 2003 twFTP – Serve Baseline.inf

```
=====
; (c) Microsoft Corporation 1997-2003
;
; Security Configuration Template for Security Configuration Editor
;
; Template Name:      High Security - Member Server Baseline.inf
; Template Version:   1.0
;
; This Security Configuration Template provides settings to support the
; Windows Server 2003 Member Server Baseline settings for the Windows
; Server 2003 Security Guide. Please read the entire guide before using
; this template.
;
; Release History
; 0001 - Original April 23, 2003
;
; * Modified version for Health Care Organization - Tumbleweed FTP/SSL Servers *
```

[Profile Description]

Baseline template for all Member Servers in an environment with high security requirements.

[Unicode]

Unicode=yes

[Version]

signature="\$CHICAGO\$"
Revision=1

[Event Audit]

AuditSystemEvents = 1
AuditLogonEvents = 3
AuditObjectAccess = 3
AuditPrivilegeUse = 3
AuditPolicyChange = 1
AuditAccountManage = 3
AuditProcessTracking = 0
AuditDSAccess = 3
AuditAccountLogon = 3

[System Access]

EnableGuestAccount = 0

[System Log]

MaximumLogSize = 16384
AuditLogRetentionPeriod = 0
RestrictGuestAccess = 1

[Security Log]

MaximumLogSize = 81920
AuditLogRetentionPeriod = 0

RestrictGuestAccess = 1

[Application Log]

MaximumLogSize = 16384

AuditLogRetentionPeriod = 0

RestrictGuestAccess = 1

[Service General Setting]

"ALG",4,"D:(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;SY)(A;;CCLCSWLOCRRRC;;;IU)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"AppMgmt",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"aspnet_state",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"CertSvc",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"NWCWorkstation",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"ClusSvc",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)"

"DHCPserver",4,"D:(A;;CCLCSWLOCRRRC;;;IU)(A;;GA;;;BA)(A;;GA;;;SY)S:(AU;FA;

| MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\ShutdownWithoutLogon=4,0

MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\UndockWithoutLogon=4,0

MACHINE\Software\Policies\Microsoft\Cryptography\ForceKeyProtection=4,2

MACHINE\System\CurrentControlSet\Control\Lsa\AuditBaseObjects=4,0

MACHINE\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail=4,1

MACHINE\System\CurrentControlSet\Control\Lsa\DisableDomainCreds=4,1

MACHINE\System\CurrentControlSet\Control\Lsa\EveryoneIncludesAnonymous=4,0

| [Privilege Rights]

SeInteractiveLogonRight = *S-1-5-32-547,*S-1-5-32-551,*S-1-5-32-544

SeRemoteInteractiveLogonRight = *S-1-5-32-544

SeDebugPrivilege =

SeDenyNetworkLogonRight = *S-1-5-7,*S-1-5-32-546

SeDenyBatchLogonRight = *S-1-5-32-546

SeDenyRemoteInteractiveLogonRight = *S-1-5-32-546

SeRestorePrivilege = *S-1-5-32-544

SeNetworkLogonRight = *S-1-5-32-544,*S-1-5-11

SeMachineAccountPrivilege = *S-1-5-32-544

SeSystemtimePrivilege = *S-1-5-32-544

SeProfileSingleProcessPrivilege = *S-1-5-32-544

SeShutdownPrivilege = *S-1-5-32-544

SeTcbPrivilege =

SeIncreaseQuotaPrivilege = *S-1-5-20,*S-1-5-19,*S-1-5-32-544

SeCreatePagefilePrivilege = *S-1-5-32-544

SeCreateTokenPrivilege =

SeCreatePermanentPrivilege =

SeImpersonatePrivilege = *S-1-5-20,*S-1-5-19

SeIncreaseBasePriorityPrivilege = *S-1-5-32-544

SeLoadDriverPrivilege = *S-1-5-32-544

SeLockMemoryPrivilege = *S-1-5-32-544

SeBatchLogonRight =

SeSecurityPrivilege = *S-1-5-32-544

Additional Reading

HIPAA

<http://www.hipaa.org/>
<http://aspe.hhs.gov/admnsimp/index.shtml>

SSH

<http://www.openssh.com/>

SSL

<http://www.openssl.org>
<http://www.ford-hutchinson.com/~fh-1-pfh/ftps-ext.html>

Certificates

<http://www.columbia.edu/kermit/security81.html#xa3>
<http://www.columbia.edu/kermit/security81.html>
<http://www.verisign.com.au/repository/tutorial/digital/intro1.shtml>

Schneir, Bruce – “Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure”: <http://www.schneier.com/paper-pki.html>

Windows Server 2003

<http://labmice.techtarget.com/windows2003/Security/default.htm>

FTP/SSL

http://www.intranetjournal.com/articles/200208/se_08_14_02a.html
<http://www.thefreecountry.com/webmaster/freetpclients.shtml>
<http://www.ford-hutchinson.com/~fh-1-pfh/ftps-ext.html#client>