# Building Secure Exchange Gateways with HTTP Extensions

Yaacov Apelbaum
Western Union

September 14, 2005

Yaacov Apelbaum is the Principal Software Architect in the eCommerce Technology and Solutions Group. In this capacity he is responsible for the development of payment systems, BI, platform integration technologies, transactional security, B2B, eCommerce, and web based decision-support systems.

## Applicable Technology:

Operating Systems: Windows 2000 and 2003 servers
Web Servers: Microsoft Internet Information Services (IIS) version 5.0 and 6.0
Web Browsers: Internet Explorer (I.E) version 5.0 and greater

## Abstract

The Windows family of servers with their integrated services leverages many of the latest Web technologies, including; RFC-compliant implementation of DAV,[1] enhanced security through strong encryption and use of certificates. Specifically, Windows 2000-2003, IIS 5.0-6.0, and ASP 3.0-4.0 offer built-in features that can be used to rapidly assemble information exchange gateways over secure and encrypted connections. This type of a solution can provide the most secure environment for synchronous transmission of sensitive data to and from the enterprise utilizing standards as stringent as SET.[2] This white paper was written to provide the reader with the theoretical and practical details on how to implement this corporate Internet gateway solution. The techniques described in this article can be implemented on any hybrid environment comprised of the Windows 2000-2003family of products.[3]

## Preface

This white paper[4] will predominantly cover Active Server Pages technology (ASP), Internet Information Server 5.0 (IIS), and the Windows NT file system (NTFS). The discussion will utilize examples written in HTML, ASP, and Visual C++ 6.0. However, this document is not designed to be used as specific implementation guide, but rather as roadmap for general technology use. For specific step-by-step instructions for installation, configuration and setup, please consult the Windows 2000-2003 Server Documentation.[5] All components covered in this review are interchangeable throughout the various versions of the operating system. Their assembly will yield a similar solution, so you can safely mix and match the environments and tools of your choice to achieve required functionality.

[1] Distributed Authoring and Versioning (DAV) is a set of HTTP extensions designed allow collaborative authoring of resources that are not necessarily stored in a file system

[2] Secure Electronic Transport (SET) This is a secure message protocol for credit card transaction handling. Visa and MasterCard, with contributions from Microsoft, IBM, GTE, Netscape, and others are currently developing this standard. SET will provide authentication for cardholders, merchants, and other users, and preserves the confidentiality of actual payment data without encrypting other non-confidential information. Unlike other secure channel communications like SSL, SET uses 56-bit Data Encryption Standard (DES) encryption, and requires digital signatures to verify the identities of all parties.

[3] For more information on these products visit: http://microsoft.com/windows2000/server/default.asp

[4] Editor's note: Portions of this white paper are quoted directly from the MSDN documentation and Platform SDK. These sections of text have been incorporated directly into the overall narration without quotes or special "Notes" marks, so as to make the text flow easier.

[5] You can also view it on-line at: http://www.microsoft.com/windows2000/techinfo/proddoc/default.asp or visit the http://msdn.microsoft.com/howto/default.asp site for many "how to" articles.

## Architectural Overview

The development of data exchange projects tends to be costly, lengthy, and complex. Common practices offer many options that can span the range of utilizing commercial of the shelf solutions (COTS) to crafting every component from scratch. The utilization of rapid application assembly techniques can in some cases be a viable alternative to the build vs. buy decision. The emphasis of this paper is on the use and of core OScomponents in building scalable and high performance enterprise solutions.

## Introduction

Integrating the enterprise with its multi faceted internal platfroms has traditionally been difficult. Integrating it with external customers and suppliers is even harder. Historically, businesses have purchased thier enterprise applications from different vendors over many years of operation, resulting in a variety of applications running on different platforms. Because each application was designed to fulfill specific tasks, such as inventory control or customer relationship management, integration with other applications was not needed or even desired. As a result, companies that decided to integrate their internal applications with external clients find the task to be cost prohibitive, difficult, and time-consuming.

The increased popularity of Internet connectivity technologies is a strong catalyst for the commerical sharing of information. This does not only apply to integration and dissemination of internal corporate data via new UI presentation models, but also to on-line collaborative initiatives such as; supply chains integration, logistics planning, team collaboration, and cross corporate data exchanges. Bridging the gap between customer needs and corporate capabilities requires the ability to respond preferably in real-time. This means implementing solutions that permit the enterprise to build seamless connections between internal back-end data and external systems and create a transparent and flexible information infrastructure that works as a tightly integrated whole.

Stiff market competition and the increased pace of technology change are putting ever increasing demands for faster enterprise connectivity and information sharing. The ability to rapidly deploy highly adaptive and cost effective business gateways is guaranteed to impact revenue generation, business agility, as well as help gain strategic advantage in the marketplace. An exchanges solution that allows partners to become productive with little or no downtime or additional investment in hardware or software, is an excllent business proposition and a marketing tool that can help strengthen the relationships and lower the overall trading costs normally associated with the traditional customer boarding lifecycle.

# Table of Contents

## Why use a Gateway?

The development of a gateway rich with functionality makes it easy to consolidate the delivery of important corporate data. An exchange gateway can enable business users to rapidly create project-based work areas that facilitate seamless collaboration between cross-organization and customer teams, which is the foundation for building a strong electronic partnership.

It is obvious that gateways should be easy to use and navigate. After all, businesses don't want to make it unnecessarily difficult for partners to interact. One way to make the gateway easier to use is to ensure that we implement familiar metaphors for common tasks. This means, storing and displaying information in consistent ways (i.e. take advantage of the Windows interface) and by using metaphors that make it easier for non-computer experts to implement the gateway and understand how it works.

There are many factors in the design of a gateway that can affect its performance, but, at the lowest level, the implementation should strive to satisfy the following key Electronic Data Interchange (EDI) objectives:[6]

- Reduce the complexity and time required to build, deploy, and maintain a proprietary solution.
- Encourage collaborative practices, provide partners with rapid information discovery, and dynamic data integration
- Allow partners to access, distribute, and share information
- Give the business partners the ability to control the operation of their information through the integrated and secure upload mechanism.
- Enables the businesses to close the loop with customers and partners by giving them the decision support processes and data they need in real-time.
- Allow for highly customizable, personalized and relevant content to be sent and received to and from customers.
- Optimize and enhance the customer experience through delivering high quality content
- Encourage business partners to do business by simplifying and eliminating implementation bureaucracy and procedure
- Forge tighter partner relationships by reducing the overall cost of implementation
- Streamline business processes and create business solutions quickly and modify them just as quickly to suit changes in market place
- Control new development costs by leveraging and squeezing every potential benefit of the existing technology, especially, proven “built-in” and freely available technology and pass the savings to the customers

## Gateway Benefits, Something for Everyone

A key factor in attracting partners is the enterprise’s ability to publish to and consume information from its partners. In this regard, electronic gateways act as an extension of the organization, they enhance streamlining, and support outsourcing arrangements that in turn reduce overall operational costs. It is evident from today’s battered economy and diminishing profits that there is the urgent justification to develop a portable gateway framework that can be easily implemented throughout the enterprise.

The ability to rapidly implement a global exchange gateway also gives the enterprise with an opportunity to differentiate itself from the competition. It also helps makes it more agile by providing an efficient way to enter new markets and responds to market changes in a timely manner. The most effective approach to

[6] Read more about the EDI (Electronic Data Interchange) model in: *Electronic Commerce, Third Edition Charles River Media, inc. 2001* – Chapter 11

enterprise gateway implementation also must ensure that the initial building blocks are Internet ready and that the implemented solution can fully exploit the platform's flexibility while providing global reach and on-demand communication capabilities. In the long run, the approach based on utilization of existing technology encourages the corporate IT to function as system integrators and further promotes solving business, rather then setup and configuration problems. A complete out-of-the-box scalable solution helps lower the overall IT costs associated with knowledge management. With improved manageability, reliability, and security, organizations can realize increased returns on their information technology investments.

Whether we like it or not, business enterprise integration is the order of the day. The need to tightly integrate with customers, partners, remote users, and the public via intranets and extranets is no longer an intellectual debate.

Without an effective exchange architecture, the enterprise will eventually lose its ability to share mission critical business information that resides on its off-host and mainframe systems.

Without a fluid multi directional communication in real time the information (see **Figure 1)** owned by the organization ceases to be manageable and becomes a maze of inaccessible data sources. On the other hand, when properly implemented, this type of a host-to-Web-to-host solution can breathe new life into legacy enterprise systems, rapidly invigorating and enhancing its ability to supply needed information to both internal and external users.

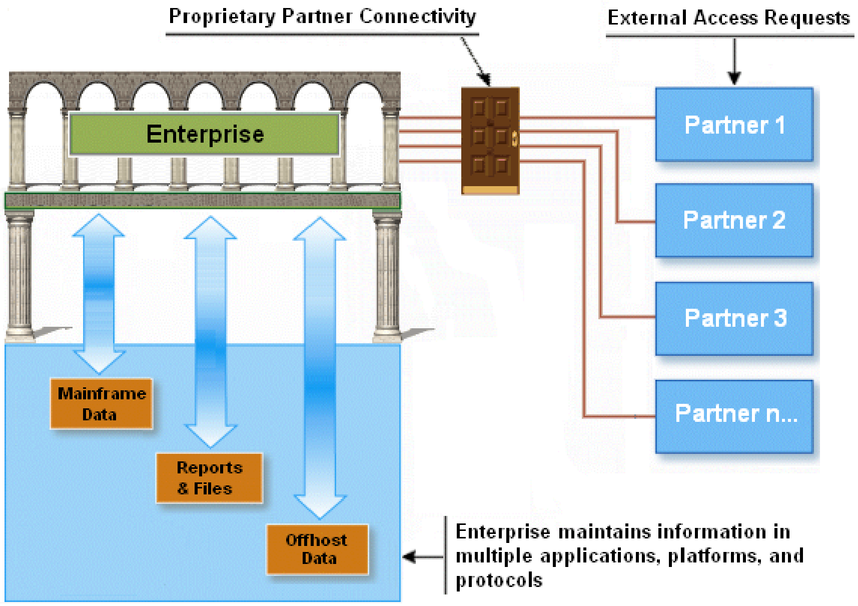

Proprietary Partner Connectivity
External Access Requests
Enterprise
Partner 1
Partner 2
Partner 3
Partner n...
Mainframe Data
Reports & Files
Offhost Data
Enterprise maintains information in multiple applications, platforms, and protocols

**Figure 1: Client's access prospective**

## Integration via Exchange Gateway

The business justification for the design of web gateways capable of supporting multiple concurrent disparate users has in recent years become the norm rather than the exception. Unfortunately, the assembly and construction of this kind of a communication and exchange mechanism is often complicated by the fact that resources (files, documents, and other raw data) are created by various divisional corporate entities, these entities as seen in **Figure 1** are scattered throughout the maze of corporate-networkdoms and lack the ability to exchange information with anyone via a structured mechanism.

Most organizations address this challenge by implementing expensive and proprietary technologies, such as dedicated connectivity solutions (mainly NDM, Dial-up, Frame Relay or Virtual Private Networks) or by creating a glut of specialty web sites designed to support specific business requirements. Unfortunately, both the traditional and the alternative solutions require a large investment in initial setup, configuration, and ongoing maintenance.

Most traditional[7] connectivity solutions are designed to support a one-to-one relationship and are not able to support multiple points of entry into the enterprise (see Figure 2). From the financial point of view, specifically, return on investment, they are only justified when the actual volume of transactions is consistently high. Operationally, and logistically, these solutions also tend to be difficult to adapt to new business requirements and are prohibitively complex to implement for smaller organizations lacking a large IT infrastructure or resources. Also, because of their proprietary design, they fail to support new and emerging technologies such as Web Services and SOAP.[8]

Under close examination, some of the cure alternatives for these traditional connectivity challenges appear to be worse than the disease. This is evident in the case of physical mailings (CD ROM or other media), e-mail, or FTP transfers. Physical mailing tends to increase the overall cost of the business cycle, encourages reactive practices (loss of batch control, introduction of new data errors, and inability to scale to large volume of transactions) as well as carrying a hefty, manual processing penalty. E-mail is inherently insecure, its delivery is rarely guaranteed, and it also it requires constant monitoring and manual intervention. FTP has been excluded by organizations as a cross-corporate communication solution on the grounds of being insufficiently secure and being susceptible to a broad base attack by a potential intruder.[9]

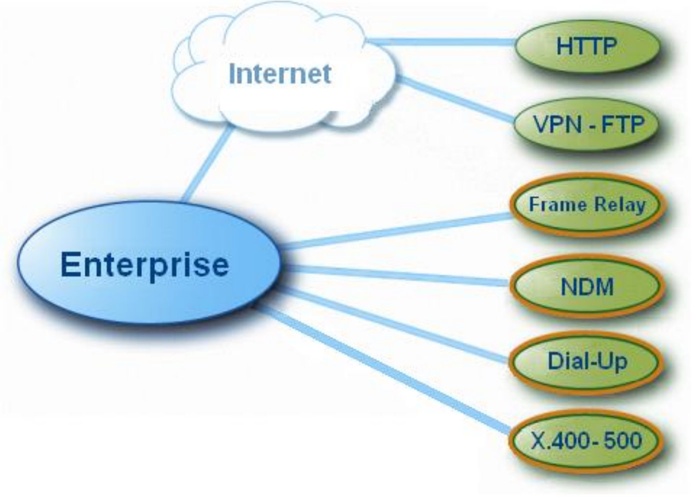


Figure 2: Traditional connectivity solutions

# The Shortcomings of the Common Design Model

While considering the challenges of creating a gateway that provides access to both legacy and off-host applications to the Web, several design patterns can be evaluated. One pattern, the Model-View-Controller (MVC) [10] used in traditional website designs initially seems to be an excellent choice for solving the gateway design challenge. It is mainly because it decouples the UI from the logic and data used to create it. However, closer examination shows the MVC design pattern uses a central controller and model to manipulate various views. In our design pattern there are multiple data objects (files and documents), operating system components (users and privileges) and a multiple views (multiple Web pages). Although the MVC pattern is ideal for a highly

[7] Only the solutions marked in orange ellipses are currently supported by FDMS

[8] A new emerging communication architecture that allows programs written in different languages on different platforms to communicate with each other in a standards-based way over HTTP and XML.

[9] Only clear-text authentication is supported with the FTP server. Because of this, having an FTP site on the Internet that uses user names and passwords could compromise your intranet's security. FTP sites on the Internet usually allow anonymous access only.

[10] You can read more about MVC at: http://atddoc.cern.ch/Atlas/Notes/004/Note004-7.html

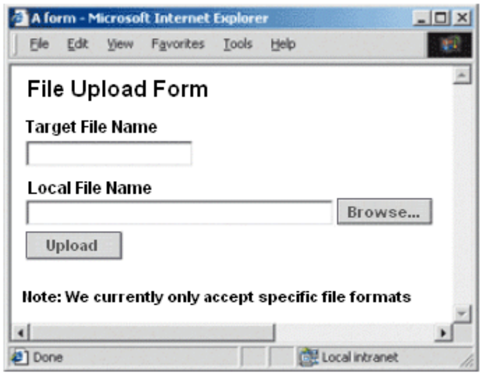
customizable data or document web applications, it is not applicable to a Web-based gateway that must support automatic configuration of user and interface components. It seems that, for an external interface, the Bridge Pattern is ideal because it defines a specific interface that is capable of connecting dynamically to a vague object such as new-dedicated user and directories that become available as the system expands.

## Looks Simple! Look Again

I have firsthand experience with implementation of custom Web based file exchange solutions (specifically file to server upload functionality) that become obsolete as soon as they were designed. This is almost always attributed to the rapidly changing security and business requirements originating from both, the customers and the enterprise.

The most common scheme for an Internet customer file upload solution is the utilization of HTML forms. In this solution, the HTML upload form uses multipart-form-data encoding to upload a file to a target web server. Typically, the ACTION attributes of the form must point to a target processor (i.e. CGI program or ISAPI Extension DLL or a Posting Acceptor) that knows how to parse the multipart encoding, process the data, and deliver it to its final destination.

A very simple type of a File Uploading page is shown in **Figure 3**. Functionality wise, the user presses the Browse… button, selects a file from either a local or remote file systems and then presses the Upload button to send the file to the server. As far as the HTML is concerned life could not be any simpler.



File Upload Form

Note: We currently only accept specific file formats

**Figure 3: File upload via HTML form**

On the server side however, you must prepare to enter development hell. Regardless of the web server or language you use, the server-side processing could not be more complicated. You need to write customized code and compile it as an ISAPI DLL [11] or CGI program [12] that will process the request. Your program will have to, among other things, be loaded and unloaded from memory, mange its own security, parse the incoming data stream, format it, generate errors, communicate with the client, create html pages, and much more. In the IIS world this process typically follows the steps seen in **Figure 4**.

These steps may include: The DLL first reads all the incoming data, it parses it, and then it puts it into some sort of table structure. Once the information is on the server, the DLL would use the data in an application-specific way, like store the file in the file system or perform other custom actions on it. Only after the DLL completes all its work on the uploaded file, it becomes available for to the user. Regardless of the functionality, I think that you get the idea. This is a laborious process that has to be manually and individually implemented for each upload solution.

[11] Windows NT Server relies on ISAPI instead of CGI as a method to efficiently request information from other applications. Unlike CGI, an ISAPI application exists as a dynamic-link library rather than an executable program. The ISAPI DLL is a communication pipe between Internet Information Server and an Internet service. The WWW service loads ISAPI DLLs when needed.

[12] The Common Gateway Interface is one method that a Web browser can use to run an application on your server. You can write a CGI application in a script and run it through a Perl interpreter; or you can compile the application in C or C++.

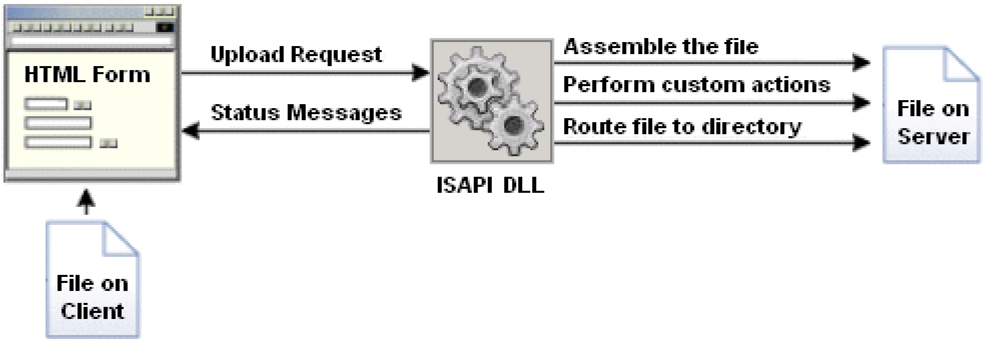


**Figure 4: Traditional web file upload architecture**

The HTML file upload example is a good illustration of how attempting to solve a straightforward business requirement can potentially bring the whole house down. It is exactly because of this reason that a good gateway solution should avoid the temptation to engage in an "in-house" development venture and, instead, utilize a solution that can be assembled from existing commercial grade components, many of which ship with Windows 2000-2003 Server.

The constantly changing and even contradictory business requirements can make any implementation of an exchange gateway obsolete even before these solutions can add to the bottom line. Most gateway applications consist of a core of standard features (user login, user reports, file upload and download functionality, etc) this collection of features that narrowly target specific user groups and requirements are always prime candidates for expensive rewrites and new bugs.

This is mainly because application features are independent of the number of users who use them. Creating narrowly specific features like customized file upload functionality through the corporate programming team may actually increase the time to complete any specialty development of an exchange gateway application. The cycle of building and fixing features until they are "right" is unacceptable in today's rapid business collaboration initiatives, where the user usually expects the product to be delivered fully matured.

## Rapid Assembly is better the Rapid Development

The ability to develop and deploy a business gateway rapidly without requiring reinvention of existing technology is the paradigm shift in assembling modern software solutions. In this development model, it is preferable to utilize the Operating System's integrated key services (such as networking, security, user management, etc.) and to expose and leverage these services to the customer. The fundamental assumption is, that these types of components have reached the highest level of maturity through the gradual product evolution, and are superior in quality and performance to any tool that can be developed in house or assembled from 3rd party components.

From my personal experience, no matter how many features end up in the final product, users will always want more. Instead of being trapped in a scenario where basic features are overlooked and omitted from the initial solution, a design based on an out of the box gateway approach is almost guaranteed to satisfy the broadest base of feature requirements because of the large feature common denominator built into Windows 2000-2003 Server. As far as addition of future custom features to the base solution, this can then be easily achieved via the standard Windows Shell extensions interface.

By leveraging an out of the box gateway solution, the IT can cut development time substantially without losing control over the quality of the application. Once a model gateway application is in place it can then be integrated with other custom made applications through the standard interfaces it exposes. Final development of the specialized features can occur in parallel as in this decentralized solution application developers take responsibility for their own features. Since the final application is ready and accepted by users faster, IT can enjoy the return on its investment for a longer period.

# Overcoming Complexity

The days of solving problems with monolithic applications are long gone. Most real-world systems these days are built by harnessing individual applications and data in ways dictated by a business process. Many of the applications we want to use will be legacy applications. Often, legacy applications become so entwined with the business that they are irreplaceable. As far as the enterprise is concerted legacy represent proven code and stability, however without few exceptions it also means no intuitive interaction via the Internet because of authentication and security concerns.

Browser based client server connectivity over the Internet requires different design and security models than the one utilized on a private network. Currently the industry provides several "typical" vendor specific solutions for exchange gateways. These solutions range from elaborate dedicated information vaults (safety deposit box) to simple secure URL re-routes to internal resources. An assessment of many of these solutions reveals that they are custom tailored to a specific market niches and only provide a limited functionality. Even in the case of customizable off-the-shelf packages, the existing plethora of 3rd party products only targets very specific business models and requirements. [13]

A disciplined approach is critical when creating the right business solution on time, in scope, and within budget. Most technology projects depend on management, business objectives, and development processes as much as on quality code. Choosing which technologies to use is only part of the effort; you also have to maximize their effectiveness. MSF will help guide you to a successful solution.

# Client Server based Gateway

In a typical implementation of a client server application, the primary design paradigm revolves around the notion of a physical permanent user connection. The server uses this one-to-one mapping from users to server to track user sessions and session-related state (i.e. transaction state). Web applications by design behave differently. It is because of the basic concept associated with the management of the unknown user (credentials and privilege level) that Web applications fundamentally differ from most typical client-server applications.

In a browser based solution, when the user directs the browser to a given Web site, the browser first establishes a TCP connection with the corresponding Web server then, the browser sends an HTTP request using the existing channel. The Web server processes the request, sends the response back to the client, and closes the TCP connection. This same process is used for every HTTP request made by the browser. The implementation of this protocol is drastically different from the client-server model where the connection is made once and used for the lifetime of the session. With HTTP, the user typically connects and disconnects on every request. This tends to complicate a gateway solution implemented solely as an HTML based website.[14]

# HTML based Gateway

The typical HTML implementation of an Internet exchange gateway typically suffers from several fundamental shortcomings that most to be addressed. Some of these concerns are:

- **High Cost -** Gateways written in HTML typically require hefty investment of financial and human resource in their development and maintenance; this is even more so, when the web site is designed to be accessed via the Internet by more than one partner. The Internet as opposed to an intranet website requires more stringent attention to security, as well as traffic and transfer volume considerations.

[13] Typical gateways are usually either Enterprise Project Management solutions used for aligning people with systems resources or project collaboration and analysis tools like CPT, or Intranets Web-based team collaborative applications used for information management and discovery like One-World

[14] Even though HTTP 1.1 offers a new mechanism called persistent connections to optimize the connection process, it still requires considerable of house keeping and management. With persistent connections, the server maintains the connection with the client for a period of time so the client can reuse the connection for subsequent requests. The client can then programmatically also extends the session via built in mechanisms to avoid timeouts.

- **GUI Richness** - Rich user interface or a user interface with a large amount of content, a complex layout, and rich user interaction can be difficult and tedious to create with HTML implementation. It is especially hard to create a rich user interface for applications likely to run in many different browsers.
- **Separation** - In a Web application, the client (browser) and server are different programs often running on different computers (and even on different operating systems). Consequently, the two halves of the application share very little information; they can communicate, but typically only exchange small chunks of simple HTTP information.
- **Stateless Execution**. When a Web server receives a request for a page, it finds the page, processes it, sends it to the browser, and then, effectively, discards all page information. If the user requests the same page again, the server repeats the entire sequence, reprocessing the page from scratch. These inherit server design means that servers have no memory of pages that they have processed. Therefore, if an application needs to maintain information about a page, this becomes a problem that has to be solved with expensive code.
- **Data Access -** Reading from and writing to a data source in traditional Web applications can be complicated and resource-intensive.
- **File Size Limit –** Large File uploads are not commonly supported by HTTP. This functionality most be implemented via expensive custom listeners and handlers on the server.
- **Security -** Server hardening typically eliminates "superfluous" server functionality, such as; scripting, listeners and default settings. This is almost guaranteed to adversely affect the site's functionality because many sites implementations relay on default server settings and presence of components or settings like scripting and access privileges.
- **Multi Access -** Gateways designed to support multiple user logins with alliance demarcation requirements (i.e. anti-trust) typically require sophisticated presentation layer that is custom tailored to each user specifications.
- **Ease of Management -** The user profiling mechanism that controls access privileges to a website is typically housed in a database located on a separate server (it is not considered a good design practice to keep user or application database on a web server). This translates to: investment in additional software, hardware as well as database maintenance and additional interface components to allow the database profiling and administration. This practice also tends to strain existing support resources due to the need to create a dedicated help desk layer.
- **User and Element Profiling -** The process of adding new users to the website tends to be lengthy and complicated because several application areas most be touched and retested. Some of these areas may include the user profile database as well as new html, ASP, COM objects and other scripts and application components. (Active Directory solutions, even though designed to alleviate this overhead, still require considerable amount of attention and handling).
- **Summary Presentation -** The HTML report page interface does not easily lend itself to the presentation of large amounts of detailed information. A user interested in viewing large number of records (i.e. report with 70,000 records) may find himself overwhelmed by the number of HTML pages returned. Also, the need to create hundreds of html pages tends to be taxing for both, the server and end user. On the other hand, even If a dynamic generation of downloadable files (i.e. creation of spreadsheet files) is implemented, the overhead involved in creating this type of file is considerable and the user is still stock with a document that is difficult to manipulate.
- **Authentication -** Due to the nature of password authentication (simple html validation of User ID and Password) associated with most websites. Simple password exploitation attack may enable a malicious user to gain access to the entire site.
- **Two-way Traffic -** HTML websites are not traditionally design for information upload because the regular implementation of the HTTP 1.0 stack does not support advanced HTTP verbs such as Copy Post, Put, and Delete. Also, most file transfer designs implemented via the native FTP service (either anonymous or logged) are not always possible because many corporate security guidelines exclude the use of plain FTP file transfers from their enterprise architecture

- **Alternative Methods -** XML and CGI date exchange requires a dedicated solution that is custom tailored to each partner. CGI even more the XML requires custom manipulation and extensive investment in coding.

## Client Session and Security

Since most web applications do not support the concept of persistent connections, there is no clear-cut method to keep track of user sessions. HTTP is a somewhat of a stateless protocol (the server doesn't remember anything about previous HTTP requests) this presents a design challenge to any gateway that most support multiple users and sessions.

It is due to fact that the Internet based architecture is connectionless and sessionless by nature; Web applications are required to implement higher-level session and security management. The browser and the server must agree on a mechanism for identifying and validating users on a connection-to-connection basis. Once the mechanism for identifying users across connections is established, the process of associating session state with a specific user and his credentials can begin. A second consideration is the overall connection security. In today's justifiable [15] insecure reality, most corporate guidelines require that most sites employ dual or triple login functionality, that the access to resources is compartmentalized, and that encryption be implemented in order to obfuscate transmitted data.

To further complicate things, exchange gateway designers must also deal with the possibility that the site or its clients will reside behind a firewall. Firewalls block potentially malicious traffic from reaching the corporate network. They do so by rejecting attempts to establish TCP/IP connections on unsecured ports. Unfortunately, since the firewall has no way to know whether these ports are secured, and because they generally have no knowledge of the port negotiation protocol used by a custom application, they can all together prevent this traffic from passing through.[16]

## The Proposed Solution

We designed our solution to primarily address the business problems and enhance the opportunities that an electronic collaboration site presents. Our solution will provide the enterprise and its partners with the tools, development framework, and the infrastructure they need to successfully design, develop, and deploy a robust and reliable Internet exchange Gateway site based on Windows 2000-2003 Server family of products.

## Overview of Functionality

Our gateway will allow internal users and multiple external partners to securely connect to a URL on the Internet and exchange information with the enterprise via an Internet Web Server. From the user prospective, the exchange gateway will appear as just another resource on their LAN or WAN.

The implemented gateway will provide the ability to consume and publish information through a variety of mechanism including integrated messaging or file delivery systems. The gateway wills also support; robust security, dedicated workspaces, and the ability reliably upload and download files regardless of format and size utilizing the CIFS.[17] The implantation of this solution will easily accommodate new and divergent business requirements and it will have no new significant costs associate with its maintenance after the initial setup.

The gateway will operate on any Windows 2000 Server product and will exclusively utilize the operating system's built-in functionality regarding; security, encryption, user account management, and file and directory

[15] Anyone who ever tinkered with NetMon network monitor or a Sniffer Pro knows how easy it is to intercept TCP packets submitted as clear text. For more information about NetMon see the Microsoft Systems Management Server or the Windows 2000 Resource Kit.

[16] HTTP is a protocol that generally passes unimpeded across firewalls; the HTTPS protocol (close cousin) can also be enabled to do the same. HTTP typically operates on port 80 and HTTPS on port 443. Both of these are known as well established ports.

[17] CIFS - Common Internet File System is an open, cross-platform technology based on the native file-sharing protocols built into Windows and other popular PC operating systems, and supported on dozens of other platforms.

access permissions. No additional tools or applications will need be purchased, licensed, installed or configured beyond Internet Explorer version 5.0 or higher and server and client certificates.

## How does it Work?

Our Internet Gateway consists of components that are already built-in Windows 2000-2003 Server. There are several additional components that have to be procured and configure. These are; several ASP scripts, certificates, grouping of user accounts and privileges, IIS restrictions and end-user directory structure profiled via NTFS.

The proposed solution in **Figure 3** is based on the approach that a client's UI is dynamically coupled to individual server objects. In this architecture, the client, after a successful authentication and login, access a group of dedicated directories (Upload Directory, Download Directory, Directory n… etc.) in order to download and upload information or perform some other custom functionality. This universal access method insures that no new interface has ever to be created, that there is no need to implement separate business logic, or develop new code.

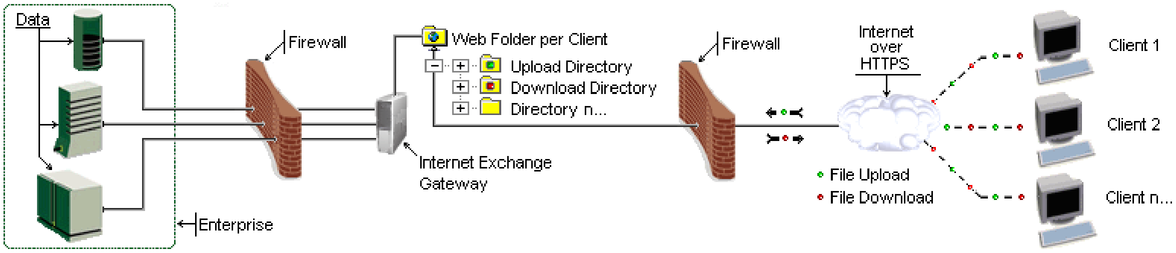
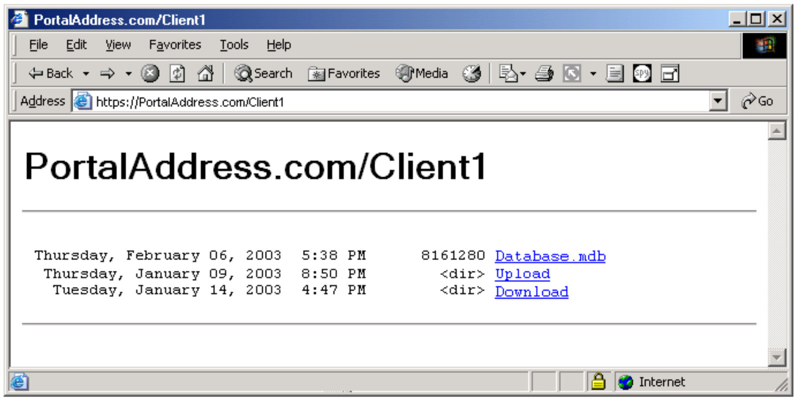


**Figure 5: Gateway's high-level architecture**

the generic directory metaphor, also allows both the enterprise and its clients to simply exchange information through a single point of interaction. Other functions that this gateway will provide: is support for directory contents synchronization, real time notification via messaging and interactive user feedback.

## Two Great Interfaces for the Price of One

The gateway users can easily integrate it without any change to into his existing workflow. This is primarily accomplishes by providing two types of built in client interfaces: a Web-based interface as seen in **Figure 6** and the Windows Explorer Shell seen in **Figure 7**.

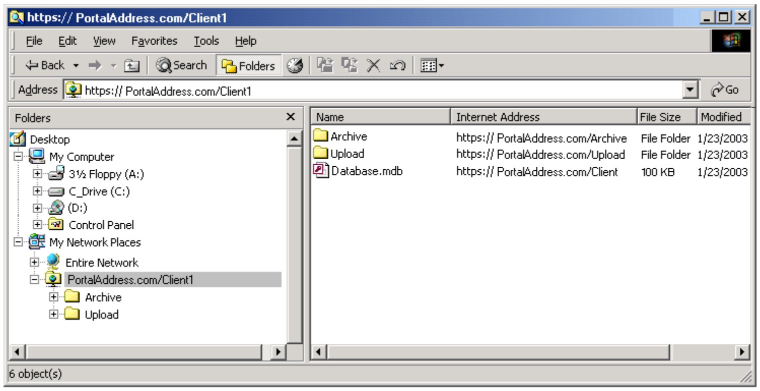


PortalAddress.com/Client1

```
Thursday, February 06, 2003   5:38 PM      8161280 Database.mdb
 Thursday, January 09, 2003   8:50 PM        <dir> Upload
  Tuesday, January 14, 2003   4:47 PM        <dir> Download
```

**Figure 6: Web-based interface**

While the Web interface provides a rich collection of GUI components to interact with, its workspaces can also be accessed through the Windows Web Folders interface. [18] Web Folders are supported on all versions of Windows, starting from Windows 95 on up. They are an implementation of WebDAV (Web Distributed Authoring and Versioning) and provide an Explorer-like view into the workspace.



| Name | Internet Address | File Size | Modified |
|---|---|---|---|
| Archive | https:// PortalAddress.com/Archive | File Folder | 1/23/2003 |
| Upload | https:// PortalAddress.com/Upload | File Folder | 1/23/2003 |
| Database.mdb | https:// PortalAddress.com/Client | 100 KB | 1/23/2003 |

**Figure 7: Windows Explorer based interface**

**Figure 7** shows an Internet Explorer based interface of the Client1's workspace. Custom functionality available in this view allows you to take advantage of the server MIME registration to display Active Documents [19] and the ability to map Web Folders to the workspace by supplying the same workspace address as the server URL. Users who use the Explorer Shell can also map a Web Folder to their workspace with an alternate alias. The Explorer view provides all the Explorer functions that users are familiar with, including the ability to easily copy, move, and delete files as well as gain programmatically access to them. Another great benefit of this interface is file association and the amount of metadata provide when you simply select a file, as you can see in **Figure 7**.

# Introduction to Internet Security

## The Mechanics of Security

I felt that the reader would greatly benefit from a general system component overview prior to examining the specific implementation scenarios. So before we get down to the nuts and bolts of our gateway architecture, let's first review each of its building blocks. I am sure, that after we complete this section, you will have a good high-level understanding of all the key concepts. I have intentionally added several low-level functionality description of some technology, this may look overwhelming, but do not despair, with a little endurance and help from the good people in Redmond there is hope.

We will start this review by examining Internet security in general and then venture into to the relationship between W2K Server and IIS security; finally, we will touch specific topics like authentication, SSL, and certificates. At the end of this white paper, there's a brief glossary of important terms and a list of resources you can use to obtain more information. I strongly encourage the reader to further investigate the more advanced topics relating to the HTTP 1.1 protocol, public key infrastructure, SSL protocol, security credentials, user authentication, and of course the Windows 2000 Server family of products. The Windows 2000 Server Resource Guide contains a wealth of information regarding this outstanding operating system. For further reading references, please consult the bibliography at the end the document.

[18] See: Microsoft article Q195851 - How to Install and Use Web Folders in Internet Explorer 5 and Q221600 - Working with Distributed Authoring and Versioning (DAV) and Web Folders for step by step installation instructions.

[19] A Windows-based, non-HTML application embedded in a browser, providing a way for the functionality of that application to be accessible from within the browser interface. For example, a Microsoft Word document would load into the browser with the full menu and toolbars.

## Internet Security, Why the Fuss?

If you have been following the news recently, you probably noticed that one of the recurring themes is the latest Internet security breach. The format of the coverage often repeats itself with boring predictability, the names of the poor victims afflicted with the misfortune, the estimated business and monitory loss they suffered, the company spokesman reassuring everyone that the problem is under control and finally a lengthy boring debate among a team of self appointed security exports that inform us that everyone in the security industry knew about this vulnerability for quite some time and how the repercussions of this attack could have been prevented, if only… If you are planning an Internet business implementation such as a gateway, especially a publicly Internet gateway, you better take a close look at how Internet Security may affect you. My mantra regarding security is; the future road to happiness and success solidly grounded on the misery and mistakes others have already experienced.

## Scope of Security

Web site security is all about managing our risks by providing adequate protections to our assets. I deliberately use the term "Adequate" because this can mean many things to different people. My definition of "Adequate" is to strike a balance between the requirements of making data freely available or entirely inaccessible. By careful partitioning of these extremes into data security requirements, it is possible to selectively implement the highest security protection mechanisms and the best price.

The accessibility to information should be driven by clear and conscious definitions such as: information's confidentiality, privacy, integrity, and availability to inside and outside users. There is no sense of investing vast resources in protecting information that is already freely available to the public; on the other hand, we should not really on ambiguity in hope that sensitive information will not be retrieved because it is kept on an obscure gateway site. The combination of security mechanisms and services, such as encryption, authentication, authorization, accountability, can all be combined to support our objectives, but their utilization should not be implemented just for the sake of making the enterprise simply "more secure".

The most important first step toward securing a web site is to analyze the business risks associated with it, the nature of systems and data to be protected, and the overall costs of the applicable security controls. We can only proceed to formulate a defense plan after determining what the most cost effective solution for the business is. In general, business web sites allowing interactive functionality justify greater levels of protection than view-only informational sites. Many business sites include multiple functions with differing security needs, so applying the highest security protections to the whole site may not be necessary and in some cases even counterproductive.

## More is not Better!

As the Internet gains deeper footing in our day-to-day life, the amount of misinformation associated with its vulnerabilities grows disproportionately. Unfortunately, there is lot of ignorant hype over how to implement site security or what is secure. These horror stories are often generated by the same security professionals that are quite whiling to hand over their "Platinum Issue - No Limit" credit cards to suspicious looking teenagers that also happen to be complete. These concerned individuals, then allow them to swipe, imprint, and copy, their credit card information without even blinking. Their defense is that all loss associated with fraud is covered by the credit card issuer. Like any business activity, the risks of using the Internet should be put in the right perspective. We currently have a comprehensive set of technologies that enable us to build secure business exchange gateways for deployment over the Internet. To perform accurate risk assessment, it is important to understand what levels of protection each one needs.

In the next section we will address the fundamental security technologies that are relevant to an Internet exchange gateway, and show how the W2K Server and related Web technologies provide the foundations for a bulletproof solution.

Our main focus will be on covering security issues in the following areas:

- Understanding and evaluate the risks faced on the Internet
- The security features available from the NT operating system and IIS
- How we can use Secure Channel Services for our transactions

## Hackers, Crackers, Safety Schmafety

It is clear that the security needs of Internet-based systems like a gateway are very much different from traditional Client-Server system operating on a dedicated network (we can implement much more stringent security on a LAN or WAN). The biggest difference between these two topologies is our inability to implement a centralized infrastructure for network security and authentication on the Internet. Also, security is further complicated because the Internet opens our application to a huge global scale, with potential user base in the range of millions.

Ironically, the initial conception and implementation of the Internet was to provide openness and robustness, and ensure the network was always available for all computers. The primary design objective for the TCP\IP protocol was to enable connectivity at the lowest connection requirements in order to survive a doomsday scenario. Even though the Internet was originally a network designed and built as part of the national defense initiative, the security and confidentiality of information was considered secondary. The designer assumed that because only trusted users where supposed to have had access to the system in the first place these trusted users would never do anything to harm the system. While most visitors to our gateway will be happy to abide by the published rules, there will always be a few “Information Perverts” who will attempt to see things never intended for public consumption. Or worse yet, even few steps further and attempt to break in, climb through, and jump over the security controls in order to damage the site.

Today's superhighway felons can be classified into the following groups:

- **The Con Artists -** They impersonate either an organization or person. For example, in a financial transaction they could impersonate the client and transfer found from an individual’s account into theirs, or in the case of an online purchase, from a web site, they could impersonate a business and use you’re the credit card number you provided them for some illegitimate purpose.
- **The Information Thief –** They intentionally access confidential information that is out of limit for them. They could for example, login to an e-mail server, download messages with strategic importance and pass them to direct competitors.
- **The Hooligan –** They usually tamper with data. For example, they could deface your web site for fun or infect a server with viruses, worms or other malware.

Of course, some of these crackers will cross exist in more the one group of affiliation. But, even more important than their affiliation with various illicit groups is that fact that many of them will originate from within the enterprise.

## Code Red! Are we are Under Attack?

Without security measures and controls in place, both inside and outside intruders might subject data to an assault. The more sensitive the data, the more likely it is that it will be attacked. Traditionally, attacks have been cataloged under two categories: *Passive* - meaning that some unauthorized information is simply monitored; and *Active* - meaning the some of the information is altered with intent to corrupt, destroy or alter the data or the network itself. **Table 3** presents some of the common attack models.

| Attack type | Attack Description |
|---|---|
| **Identity interception - Sniffing or Snooping** | The intruder discovers the user name and password of a valid user. This can occur by a variety of methods, both social and technical. |
| **Masquerading** | An unauthorized user pretends to be a valid user. For example, a user assumes the IP address of a trusted system and uses it to gain the access rights that are granted to the impersonated device or system. |
| **Replay Attack** | The intruder records a network exchange between a user and a server and plays it back at a later time to impersonate the user. |
| **Data Interception** | If data is moved across the network as plaintext, unauthorized persons can monitor and capture the data. |
| **Manipulation** | The intruder causes network packets to be modified or corrupted. Unencrypted network financial transactions are vulnerable to manipulation. Viruses can corrupt this information. |
| **Repudiation** | Network-based business and financial transactions are compromised if the recipient of the transaction cannot be certain who sent the message. |
| **Man-in-the-Middle** | Diversion of IP packets to an unintended third party, to be monitored and possibly altered. |
| **Malicious Macros** | Application-specific viruses exploit the macro language of sophisticated documents and spreadsheets. |
| **Denial of Service - DOS** | The intruder floods a server with requests that consume system resources and either crash the server or prevent useful work from being done. Crashing the server sometimes provides opportunities to penetrate the system. |
| **Mobile Malicious Code** | This term refers to malicious code running as an auto-executed ActiveX® control or Java applet downloaded from the Internet. |
| **Abuse of Privileges** | An administrator of a computing system uses full privileges over the operating system to obtain private data. |
| **Trojan and Other Horses** | This is a general term for a malicious program that masquerades as a desirable and harmless tool. For example, a screen saver that mimics a logon dialog box in order to acquire a user's name and password and then secretly sends that password to an attacker. |
| **Social Engineering** | Sometimes breaking into a network is as simple as calling new employees, telling them you are from the IT department, and asking them to verify their password for your records. |

**Table 1: Common Attack Scenarios**

# Operating System Security

## Security Must be More then Skin Deep

The close integration between the web server and the OS allows us to leverage the benefits of the robust security that is built into the very core of W2K Server. The construction of the Windows 2000-2003 Server was designed to meet the most stringent criteria of the C2 Security requirements. [20] Among other things, this criterion dictates the critical need for an operating system to be designed for optimal security from the ground up. Some of the requirements that it most meet are:

- It must be possible to control access to a resource by granting or denying access to individual users or named groups of users.
- It Memory must be protected so that its contents cannot be read after a process frees it. Similarly, a secure file system, such as NTFS, must protect deleted files from being read.

[20] C2-level security as defined by the U.S. Department of Defense

- Its Users must identify themselves in a unique manner, such as by password, when they log on. All auditable actions must identify the user performing the action.
- Its System administrators must be able to audit security-related events. However, access to the security-related events audit data must be limited to authorized administrators.
- It must be protected from attempts of unauthorized external interferences or tampering, such as modification of the running core system components or of system files stored on disk.

## Native Integration with Operating System Services

Internet Information Server (IIS) is a native service that runs on all versions of Windows 2000-2003 Servers (W2K Server) The robust security architecture of this OS is used consistently across all system components, with successful authentication controlling access to all system resources. IIS is also fully integrated into the Windows security model and it inherits all its functionality and overall permissions from this model. Because IIS uses the Windows user database, administrators do not need to create separate user account management systems for each individual website.

It is mainly because of this tight integration with the OS, that IIS can natively take advantage of many of the system tools provided by the OS. For example, we can use the built-in auditing functionality for monitoring of resource use, or we can monitor failed attempts to access secure files or directories. All these events, can be recorded in the Windows Event Log, and audited with the same tools used for managing the OS. [21]

## Strong Security and Low Overhead

Many of the leading Web servers tend to install their own security implementations on top of the target operating system. Security in this type of implementation is usually achieved via a safety container that is managed by the web server. Unfortunately, this safety container metaphor creates additional system overhead as well as potential security exposure due to lack of deep integration and proper service synchronization with the OS. This is not the case with IIS-W2K Server, files and system objects can only be accessed with the proper user permissions. User and group accounts are managed by a globally unique identification. When accounts are deleted, all access permissions and group memberships are deleted through a cascading inheritance effect. So, even if a new account is created using a previous user name, none of the permissions are inherited and the user's access to resources such as HTML pages, scripts, shared files, printers, and databases has to be re established.

## IIS and NTFS

Even though IIS can properly operate on as a system running a File Allocation Table (FAT) [22] hard drive, you should consider leveraging the format to Windows NT File System (NTFS). The following are some of the benefits:

- Contrary to FAT, NTFS is not accessible from DOS. This makes the resources more secure from attacks using DOS commands or shell.
- NTFS allows us to configure the Access Control List (ACL) to grant or deny various forms of access to user and group accounts.
- The NTFS file system is more efficient when it comes to managing hard drive space.

[21] You can use security-auditing techniques to monitor a broad range of user and Web server security activities. It is recommended that you routinely audit your server configuration to detect areas where resources may be susceptible to unauthorized access and tampering. You can use the integrated Windows utilities, or the logging features built into IIS to achieve this.
The ability to determine who's done what, which files or pages have been accessed, and what may have been compromised or tested. The two most common logs are: **Windows Event Logs -** typical show activities such as user login, hardware state, and so on. The **IIS Logs** - provides information on who is accessing your site and specifically what content they are accessed.

[22] Because this security is not available for file allocation table file systems (FAT) or FAT32 file systems, consider using NTFS for all of your Web sites to provide this additional security benefits of NTFS file and folder security.

NTFS ACLs extend the granular security that is available for Web sites. When Web sites are installed on NTFS file systems, user rights and access permissions for Web resources are controlled by file system ACLs. We can configure these lists to control access to individual Web sites, folders, or files. We can use Windows Explorer to grant or revoke rights and permissions for user accounts and security groups. When folders and files have ACL restrictions, Internet Information Services prompts users to enter their Windows user names and passwords for authenticated access, this even occurs in the extreme case when anonymous access is enabled for the requested resource.

## The IIS Security Control Model

IIS has a native built-in support for four different security control mechanisms: IP address access, user access, virtual directory access, and integrated Windows NTFS access. IIS is flexible in that it allows the Web administrator to freely implement any combination of these security mechanisms as necessary. The structure of this security mechanism is such that in case of conflicting requirements, by default, the more stringent control is implemented.[23]

The first line of defense in the IIS security model is the ability to grant or deny access to the Web server based on the IP address of the requesting client. The IP addresses of certain host machines may be specified and either granted or denied access to the Web server (this is one nice way to get rid of individual internet pests). When any packet of data is received, IIS looks at its source IP address and then checks it against those stored in the exclusion list. If it finds the IP address there, it proceeds to apply the predefined actions to it. When using IP address access control, note that some Web clients may be accessing your server through a proxy server or firewall. When this happens, the IP address of the incoming packets will be that of the proxy server or firewall itself, not of the actual user (this is common for LAN traffic as well).

One good implementation for this type of IP address access control is useful for restricting access to or from an entire domain. The current limitation of this is that once it is configured, it will govern the entire Web server and it cannot be applied on individual or virtual-directory level.

## The IIS Access Privileges

IIS defines two access privileges, *Read* and *Execute*, that can be applied to individual directories, virtual directories and all of the files contained in them. With *Read* permissions, Web clients are allowed to read or download files stored in a virtual directory or subdirectory thereof. A client request to read information in a regular or virtual directory that does not allow read access would result in an error [24] being returned to the client. Generally, only directories that contain information to be published or downloaded should have *Read* permissions enabled. To prevent clients from downloading executable files, directories that contain CGIs or DLLs should not enable *Read* permissions. Instead, these directories should have *Execute* permissions so that Web clients would be able to run them. An attempt by a client to run applications in a regular or virtual directory that does not have *Execute* permissions will also result in an error.

IIS has built-in support for four different security control mechanisms: IP address access, user access, virtual directory access, and integrated Windows NT File System (NTFS) access. IIS is flexible in that it allows an administrator to freely implement any combination of these security mechanisms as necessary. **Figure 8** shows the various security arrangements and how they relate to each other.

[23] For more on IIS related security visit: http://www.15seconds.com/focus/Security.htm

[24] IIS enables administrators to customize the message that is displayed to the client in the case of HTTP errors (such as "404 Not Found"). Instead of using the HTTP 1.1 error messages that are returned to the client by default, Web site administrators can choose to send the client other error messages. These customizable error messages can be in the form of a file or a URL.

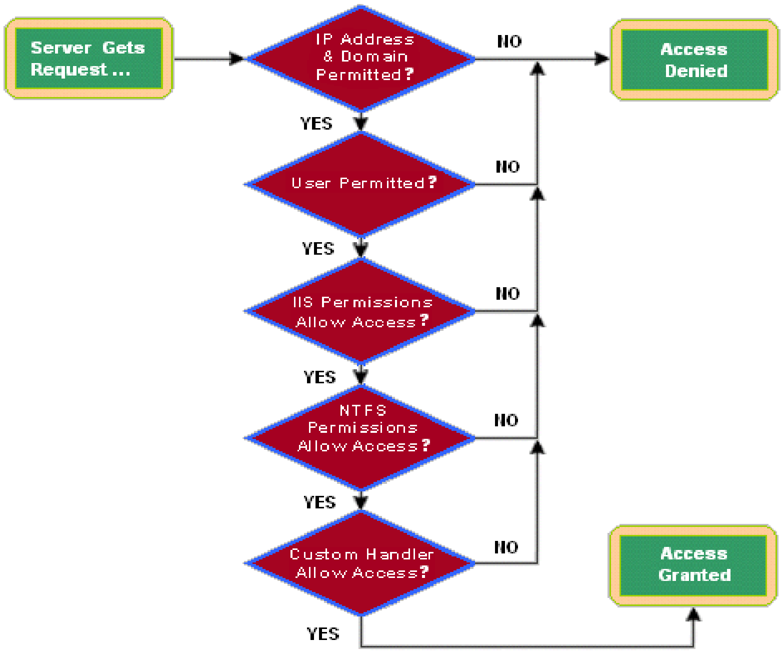

Server Gets Request ...
IP Address & Domain Permitted?
NO
Access Denied
YES
User Permitted?
NO
YES
IIS Permissions Allow Access?
NO
YES
NTFS Permissions Allow Access?
NO
YES
Custom Handler Allow Access?
NO
Access Granted
YES

**Figure 8: Access permissions process flow**

At a high level, this login process is simple. The server gets a request, it then goes through a series of validations, and then either denies or grants access based on the results of each validation step. The interesting thing about this process is that, in order to obtain access, the requester has to go through the entire chain of verification. If any single step fails at any point along the way, access is immediately denied. This is a great concept, because it significantly reduces a scenario were by an attacker could get his foot in the door by providing partial credentials and then proceed to elevate his privileges.

The second piece of the IIS security model is its integration with the security provided by the W2K Server. As stated previously, a user must have a valid account or he a member of a user group to access a W2K machine both, locally or remotely over a network. This user account is used to monitor and administer access to server resources such as files, printers and directories. Windows security is administered using the Access Control List (ACL) for each resource on the server. Using the ACL, users or groups of users are granted or denied access to one or more resources. Windows file and directory security only applies to those files and directories that reside on an NTFS partition. FAT partitions do not support ACLs. [25] Fortunately, this is not a scenario that would concern us because Windows 2000 Servers and latter, must always be installed with an NTFS file system.

File and folder ACLs for an object may be accessed and modified by locating the object using the Windows Explorer, right-clicking on it, selecting Properties, and going to the Security tab. NTFS offers five standard types of file access modes: No Access, Read, Change, Full Control, and Special Access, **Table 2** describes them. By default, the Everyone group has full control of all files created on NTFS partitions. If a conflict arises between the access allowed by NTFS and that allowed by an IIS directory, the strictest access is always applied.

| Access Type | Access Description |
|---|---|
| **No Access** | Users have no access of any type |
| **Read** | Users can view files |
| **Change** | Users can view and change files |
| **Full control** | Users can do anything to a file, including deletion |
| **Special Access** | Customizable access privileges |

**Table 2: User access types**

[25] In a FAT based file system all users are granted full access rights to all files. This makes for a very insecure server environment and this type of a configuration should be avoided at all costs on a web server.

## IIS and NTFS Permissions on Files and Folders

The NTFS level permissions are respected by IIS, so manipulating individual permissions on files and folders allows better granularity as you control what pages can be accessed by whom. A common scenario would be to have customer pages on a read site set with the default "Everyone Full Control" permissions, while other pages that allow the editing to be restricted to Administrators. A good rule to remember is that IIS can only talk about HTTP requests, not file system requests. The Permissions Mantra is: If Web and NTFS permissions do not agree, the more restrictive of the two will be used for HTTP requests.

## Introduction to Access Control

With NTFS access permissions, the foundation of the security for your Web server, you can define the level of file and directory access granted to Windows users and groups. For example, if a business decided to publish its inventory on a Web server, you would need to create a Windows user account for that business and then configure permissions for the specific Web site, directory, or file. The permissions would enable only the server administrator and the owner of the business to update the content for the Web site. Public users would be allowed to view the Web site, but not alter its contents.

The first and most fundamental area of access control is used to determine who should, or should not have, access to either the whole web site or to specific user directories on the site. For example, if you are planning on having some sort of access schema, we can restrict access by utilizing the following categories:

- **Anonymous -** Allows anyone to view the content on your site. Anonymous, Basic, and NTLM can all be set through the same IIS dialog box using the Microsoft Management Console MMC).
- **Basic -** Requires a user ID and password. Not very secure, since it is sent over the wire either as clear text or base64-encoded. Still very appropriate for some applications, and probably the most widely used authentication method.
- **Digest Authentication -** Conceptually similar to Basic; however, the password is not sent over the network. Instead, a hashed version of the password is used. This is not officially supported in IIS 4.0. However, since it is a proposed part of HTTP 1.1, I thought you might come across it. This may end up being a good method to use in the future, since it will likely be supported by multiple browsers and will get around some of the major problems of Basic Authentication.
- **NTLM (NT Challenge/Response)** - The most secure of the three basic authentication methods supported by IIS. However, you must be using Internet Explorer clients to support NTLM.
- **TCP/IP Addresses -** Allows you to restrict access based on a user's IP Address or domain. You can programmatically restrict access according to a domain as well, but that is a much more complex option, and will be addressed in detail in an upcoming column.
- **NTFS Security -** Allows you to specify permissions at the file level, based on user or W2K Server groups.
- **Site Server Membership** - Part of the Site Server product, which sits on top of NTS and IIS. You can use it when you need NT Authentication, but want higher scalability or are on the World Wide Web, where your users may not participate in an NT domain model. This is an ideal solution for large subscription scenarios.
- **Content Rating -** A self-selecting type of access control that we normally have no control over. Users must configure a response to this in their browsers.

## Groups are not Only for Groupies

At the core of W2K Server security is the User and Group account structure and its logical extensions, the user group and individual user privileges. When IIS is first installed, it creates two user accounts, assigns them specific user rights, and associates them with a specific user group. These default user account as well as other accounts that can be created manually are used by IIS to grant access to Web resources.

Generally, a group is a collection of one or more user accounts. The utilization of groups that contain individual users is an effective mechanism for granting common capabilities to a number of accounts in a single operation. This is especially useful when administrating systems with a large number of accounts like the once found in a multi user web server. The default user privileges in W2K Server are described in **Table 3**. The permissions are granted to the following base groups: [26]

| User Account | Account Privileges |
|---|---|
| **Administrators** | Members of the Administrators group can perform all functions supported by the operating system. The default security settings do not restrict administrative access to any registry or file system object. Administrators can grant themselves any rights that they do not have by default. |
| **Guests** | Users requiring temporary access to the system. |
| **Power Users** | Members of the Power Users group have more permissions than members of the Users group and fewer than members of the Administrators group. Power Users can perform any operating system task except tasks reserved for the Administrators group. |
| **Users** | The Users group provides the most secure environment in which to run programs. On a volume formatted with NTFS, the default security settings on a newly installed system are designed to prevent members of this group from compromising the integrity of the operating system and installed programs. Users cannot modify system-wide registry settings, operating system files, or program files. Users can shut down workstations, but not servers |
| **Backup Operators** | Members of the Backup Operators group can back up and restore files on the computer, regardless of any permission that protect those files. They can also log on to the computer and shut it down, but they cannot change security settings. |
| **Special Groups** | Specialty users created by IIS, or other native Windows applications. The scope and their privileges is govern by the application's functionality |
| **Print Operators** | Users that can manage printers |
| **Replicator** | The Replicator group supports directory replication functions. The only member of the Replicator group should be a domain user account used to log on the Replicator services of the domain controller. Do not add the user accounts of actual users to this group. |

**Table 3: Group access privileges**

IIS also depends upon user accounts while providing Basic and Windows Challenge-Response authentication. In order to complete successfully, both of these methods require that valid user accounts be in place. This is necessary because although IIS creates and configures the dedicated account, which is used for anonymous authentication, it does not create any accounts for Basic authentication. IIS assumes that you have created, or will create, Windows accounts for use with Basic and Challenge-Response.

Note, that you must have valid Windows accounts to use with Basic and Challenge-Response authentication. Neither Windows nor IIS will create them for you. Without such accounts these authentication methods will not work. This is done by design in order to force you to make compartmentalize the "extra secure" users typically associated with Challenge-Response connectivity.

# May I See Your Papers? Or Authentication

When a user connects to an Internet server, he may send and receive sensitive information across the line. Often, this information may include credit card numbers, social security numbers as well as other sensitive information. Many users concerned with security look for the little security icon in their browser's status bar for the validation that their online sessions are being encrypted. But what is the point of encrypting a message if you don't know whom you are communicating with? How do you know that the connection is really made to HonestUsedCarDealer.com and not to their arch evil competitor DishonestUsedCarDealer.com? What is needed is some form of authentication that scales to the global Internet.

[26] For more information on user and group account privileges see the: Local User and Group topic in the Microsoft Management Console online help documentation.

A typical implementation of a secure connection over the Internet could utilize many technologies, this can include: digital certificates, encryption, and operating system challenges. In this type of a solution the ability to prove the identity of the client and the server is of paramount importance. For example, if you are supporting on-line financial transactions, such as account balance transfers, you would make sure that the communications channel is secure, and also that whomever executed the transaction was the true owner of the account.

## Data Integrity and Privacy, or Your Right to Know

Privacy and data integrity go hand in hand; if your communication is secure, then no one should be able to alter the information that you send or receive. Privacy techniques ensure that nobody else has access to your secure communication. Note that posting from HTML forms is generally not considered a private way to transfer data that applies to HTML streaming data as well.

Data integrity methods ensure that the data you send is the data your user receives, and vice versa. Again, financial transactions are an area illustrating the risks of either corrupting or malicious alteration data. If you purchase something, you want to make sure that the amount hasn't been altered, either maliciously or through an error caused by a system crashing before the transaction was executed.

## W2K Server Challenge and Response Modes

Windows implements a challenge-response authentication scheme known as NT LAN Manager (NTLM). This authentication was originally developed by Microsoft. It was used in Microsoft LAN Manager on Windows 3.11 for Workgroups - one of Microsoft's earliest networking products.

A NTLM Challenge-response process can occur when a user or service (such yourself or IIS) tries to access any resource stored on a W2K Server machine across a network (such as when viewing a shared resource on a server). This mechanism can be used by IIS to authenticate a user browsing a Web site. Windows NT Challenge-Response is W2K's secure way of determining who is making a request (i.e. authenticating them). Windows 2000 Challenge-Response authentication uses user accounts directly, and does not transmit logon information in clear text, such as the user name and password, over the network or the Internet. This is very important if you want to safeguard the user names and passwords of our Internet clients. Also, because this type of authentication uses individual user accounts, we can have a great deal of flexibility over the access level of those clients.

Challenge-Response authentication uses hashing machinery to transmit credentials as a hashed value that is not easily readable or decipherable. During the hashing process, a copy of a plain text message is run through a mathematical procedure that results in a hash value that's usually 128 to 160 bits long. The hash value is said to be one-way, that is, it's computationally difficult to reverse engineer the original plain text message from it Challenge-Response authentication uses a series of these hashed exchanges between the server and the client to establish the identity of the user before granting access to resources. The following are the main benefits behind the utilization of hashing in establishing a secure connection:

- All hash values from a particular hashing algorithm are the same size. This means that if you take a five-word message and a complete set of encyclopedias and run both through the same hashing algorithm, the resulting value of each would be the same size. This makes a proximity attack almost impossible
- Hash values from very similar messages are very different. Let's say that you were to take a complete set of encyclopedias and run it through a hashing algorithm. Then you add one single character to one of the volumes and run it through the same hashing algorithm. The two values derived from this, other than being the same length, would not look similar. This again helps defeat a side scan and proximity attacks
- One hash value may have an almost infinite number of messages that could have produced it. This is the reason why it is computationally infeasible to derive the original message from a hash value.

Because both the client and the server are authenticating each other during the cryptographic exchange, no one can later step in and impersonate the server. This almost guaranties that users know whom they are communicating with. But just like any good things, the process does have these limitations:

- Challenge/Response does not work well on a secure extranet because it cannot operate over a proxy server or other firewall application. That said, Challenge-Response works great in intranet scenarios.
- The only browser that supports Challenge-Response is Internet Explorer, version 2.0 or higher. This may restrict the user base our application could service on the Internet if this is the only type of authentication we provide.

## Basic Authentication

Basic authentication is widely used on the Internet because it is fast and easy to implement; most common browsers support it. It also provides fair levels of security. But it could theoretically allow a determined potential hacker to intercept IP packets going and coming from client to server. [27]

Basic authentication transmits the user name and password over the wire with Base64 encoding. When using Basic authentication, it is good to keep the following things in mind:

- Utilities to decode Base64 encoded packages can be easily written and are readily available on line. This means that intercepting these packages over the Internet is relatively easy. Once the IP packets are intercepted and decoded, the password of the account being used is plainly visible.
- At no time is the user assured that they are sending their password information to the correct server. In other words, there is no server authentication with Basic authentication and a potential "IP Spoofing" attack can be initiated.
- It is possible to create programs that can "spoof" the Basic logon dialog and capture user logon information, such as their account password.

The most recent industry standard development in Web security is the Digest Authentication specification. Digest Authentication is slated to be a replacement for the Basic Authentication implementation. Basic Authentication is the present industry standard means of identifying the credentials of a user submitting a browser request. But Basic Authentication as we have just seen, has a glaring weakness in that the user ID and password are sent as clear text. Digest Authentication is intended to be a stopgap measure by the W3C to fix the security holes of the Basic Authentication mechanism.

Digest Authentication uses a hashing algorithm to form a hexadecimal representation of a combination of user name, password, the requested resource, the HTTP method, and a given randomly generated variable value that is sent with the return challenge from the server. This is only a rudimentary encryption of the password and should not be assumed to be strong form of security. Digest Authentication is not as secure as Kerberos [28] or a client-side key implementation, but it does represent a stronger form of security than Basic Authentication.

Digest Authentication is an HTTP 1.1 specification, which requires that a browser be compliant to this specification. Since a hashing function must encrypt the user name and password, the browser must handle that prior to submitting it to the server. If any IIS 5.0-6.0 virtual directory has Digest Authentication enabled, a request from a browser that is not HTTP 1.1-compliant will generate an error in the browser request.

## Making Basic Authentication Even More Secure

You may be wondering at this point; if Basic Authentication is to week and Challenge-Response is not applicable to the Internet, how am I supposed to achieve a secure connection that is supported by most browsers? The answer is to use Basic authentication encrypted by SSL. This combination will help you achieve tight access control to your sensitive information without the fear of logon information being intercepted and used maliciously. It also allows you to authenticate your server so that users can be confident about your identity.

[27] IP packets are the unit by which information is sent over networks, like a letter in an envelope.

[28] The Kerberos protocol is the protocol of choice in Windows 2000. The Kerberos v5 authentication protocol is the default for authentication of users who are logging on to domain accounts from computers that are running Windows 2000.

## Web Storage System: Storage to the Stars

W2K Server and IIS supports a new storage technology called the Web Storage System, which combines the features and functionality of the NTFS file system, the Web, and a collaboration efforts through a single location. You can use it for storing, accessing, downloading, and managing information, as well as for building and running applications. Every item in the Web Storage System is URL-addressable and fully supports semi structured data such as documents, messages, reports, HTML files, and ASP pages. The Web Storage System provides access using the HTTP and HTTPS protocols, which has been enhanced through the WebDAV specification to support an additional set of protocol commands.[29]

A Web Storage System, is a hierarchical folder system, much like a file system, that can hold documents, e-mails, Web pages, multimedia files, customer reports and server as both, upload and download portal. It can be accessed via Web browser (HTTP), file system drive mapping, Data Access Objects (such as ADO 2.5), and wireless protocols. Web Storage Systems act like a potluck dinner for corporate data. Bring whatever data you want and the Web Storage System will complement it with authentication, encryption and all the other robust built-in services that come with Windows 2000.

## Web Storage System is Wholesome

Web Distributed Authoring and Versioning (WebDAV) [30] is a set of standards-based extensions to HTTP 1.0 that enable us to share and work with server-based HTTP Web files and is implemented using well-formed XML. This capability is available regardless of the authoring tools, platforms, or types of Web servers or document management systems the documents are stored in.

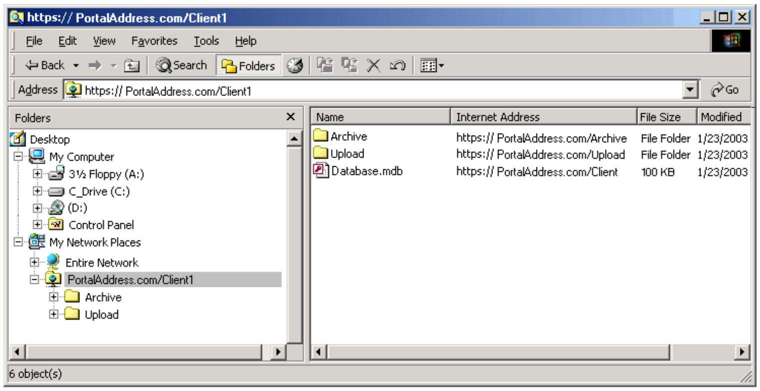
Internet Explorer 5 supports WebDAV, enabling us to navigate to a WebDAV-compliant server, such as Microsoft Internet Information Server (IIS), and view the server as if it were a part of the local file system. We can then also drag and drop files and perform other tasks, such as moving, copying, and saving files between local and remote WebDAV-compliant servers.

DAV works as a method of procedures for authors to maintain content on a Web server by providing the security structure that defines access to hierarchical elements such as directories and files. IIS implements DAV to allow remote authors to open, edit, move, search, or delete files on the Web server. The DAV specification includes additional features such as version management and access control via digest authentication, a security scheme that is also a new extension to HTTP 1.1.

Finally, DAV has also been incorporated into all Windows 32 bit operating systems through the implementation of Web folders. You can view your Web folders by simply opening the Windows Explorer typing your URL and successfully completing the login sequence. After establishing your connection, you can then access the files just as if they were on your local hard drive or network file server.

[29] For more information on Web Storage visit: http://www.ics.uci.edu/pub/ietf/webdav

[30] For more information on WebDAV visit: http://www.ietf.org/html.charters/webdav-charter.html



Address: https:// PortalAddress.com/Client1

- Desktop
  - My Computer
    - 3½ Floppy (A:)
    - C_Drive (C:)
    - (D:)
    - Control Panel
  - My Network Places
    - Entire Network
    - PortalAddress.com/Client1
      - Archive
      - Upload

| Name | Internet Address | File Size | Modified |
|---|---|---|---|
| Archive | https:// PortalAddress.com/Archive | File Folder | 1/23/2003 |
| Upload | https:// PortalAddress.com/Upload | File Folder | 1/23/2003 |
| Database.mdb | https:// PortalAddress.com/Client | 100 KB | 1/23/2003 |

6 object(s)

**Figure 9: Accessing a Web Server Directory**

## Revisiting Web Folders

The Web Folder behaviors available Internet Explorer (IE) 5.0 and higher allows users to navigate to a folder view, and include support for Distributed Authoring and Versioning (DAV) and Web Extender Client (WEC) protocols. DAV is a series of extensions to the http and defines how basic file functions, such as copy, move, delete, and create folder, are performed across http. WEC is a Microsoft FrontPage® protocol that provides the same sort of functionality as DAV, in addition to its own value-added FrontPage features. Both protocols define how to set and retrieve properties on http resources.

The Web Folder Behaviors enable authors to view sites in a Web folder view, which is similar to the IE folder view. The DAV and WEC protocols add additional capabilities to the Web folder view. For example, using the Web Folder Behaviors and DAV makes it possible to perform the equivalent of a DIR command on an http resource and retrieve all the information necessary to fill a Windows Explorer view. Also, as we already indicated earlier, another great benefit derived from this implementation is the browser's supports for two Web Folder behaviors: the Internet Browser view and the Windows Explorer view.

The Web Folder behaviors encompass the means to navigate from a Web page to a Web folder view. They provide convenient access to folders and files on an http server. A Web folder view maintains a consistent look and feel between navigating the local file system, a networked drive, and an Internet Web site. Although a Web folder is a part of the file system hierarchy, it does not necessarily represent anything in the file system.

Web Folders is a Web authoring component that is included with Internet Explorer 5. When you use this component, you can manage files on a WWW Distributed Authoring and Versioning (WebDAV) server by using a familiar IE Browser or Windows Explorer interface. WebDAV is well known extension to Hypertext Transfer Protocol (HTTP) version 1.1 that defines how basic file functions such as copy, move, delete, and create folder are performed.

## Benefits of Web Folder Behaviors

The Web Folder Behaviors provide an easy way to view files and folders on an http server for anyone using Internet Explorer 5 or later. In addition, script authors are empowered with a robust implementation that exposes any HTML object to the Web folder navigation abilities of the Web Folder Behaviors, while delivering to authors the extensibility for designing their own interface. Some of the advantages of using Web Folder Behaviors are:

- Anyone with Internet Explorer 5 can open a Web folder with no script and no fuss.
- Internet service providers (ISPs) can provide Web page clients with easy file and folder access to their content.
- Authors can access their content in Web folder view while maintaining control over the look and handling of the Web Folder Behaviors

## How to Use Web Folders to Manage Files

The specification in RFC 1867 "Form-based File Upload in HTML" describes the mechanism by which a file may be uploaded from a Web browser to the server. However, to implement this functionality it is necessary to install the Posting Acceptor bundled with IIS. However, this native posting acceptor cannot be used for generic functionality because of its inability to process large files and handle extensive errors.

Web Folders installs as a namespace or shell extension with an icon in My Computer (root object in Windows Explorer). This root object is a container for shortcuts to your Web publishing sites. You can use Windows Explorer to view, move, copy, rename, delete, create new, sort or group files by properties, and view property sheet information for files in a Web folder, depending on your authoring and security permissions on the Web server.

## Gateway Implementation Overview

Our proposed gateway is a powerful application that will allow us to build a secure upload and download server over the Internet. The site will support the upload of documents and files, such as Microsoft Word documents, text files, HTML documents, Microsoft Excel spreadsheets, binary files and so on.

We will accomplish this Using Active Search Page (ASP) redirector page, user dedicated directory structure with specific permission, and local user access groups. Upon successful login, the site visitor will be routed directly to his work area and given stratified access to the various resources located there. By clicking on the file name (depending on his interface of choice) the user will either display the document in his browser or proceed to download it to his workstation.

## Conclusion

For enterprises large and small, the features of an Internet Exchange Gateway provide a powerful yet flexible solution for users to easily and quickly find and exchange enterprise information.

In summary, the Exchange Gateway provides the following benefits:

- Personalization
- Multi tier security via OS based ACL, NTFS
- Support for large user population via Active Directory integration
- Push Pull Support
- OS API provides support for a high degree of automation and scripting
- Excellent platform for Web 2.0 style knowledge management applications
- Secure Internet access
- Good integration with none Microsoft Windows based servers and Web Servers

# Appendix

## Microsoft Knowledge Base Articles (http://support.microsoft.com)

| | |
|---|---|
| **Q239120** | Create a Secure FTP Directory that Uses Password Authentication |
| **Q299970** | Ho to use NTFS Security to Protect a Web Page Running on IIS 5.0 |
| **Q266118** | How to Restore the Default NTFS Permissions for Windows 2000 |
| **Q164882** | Practical Recommendations for Securing Internet-Connected Windows NT Systems |
| **Q282060** | Resources for Securing Internet Information Services |
| **Q271071** | Minimum NTFS Permissions Required for IIS 5.0 to Work |
| **Q229694** | How to Use the IIS Security "What If" Tool |
| **Q299875** | How to Implement SSL on a Windows 2000 IIS 5.0 Computer |
| **Q298805** | How to Enable SSL for Customers Who Interact w/ Your Web Site |
| **Q201771** | How to Set Up an FTP Site So That Users Log Onto Their Folders |
| **Q164015** | Understanding TCP/IP Addressing and Subnetting Basics |

## Requests For Comments (RFC) (http://www.ietf.org/rfc)

| | |
|---|---|
| **2459** | Internet X.509 Public Key Infrastructure Certificate and CRL |
| **2660** | The Secure Hypertext Transfer Protocol. |
| **2518** | HTTP Extensions for Distributed Authoring |
| **3281** | An Internet Attribute Certificate Profile for Authorization |
| **0172** | The File Transfer Protocol |
| **0265** | The File Transfer Protocol |
| **0265** | The File Transfer Protocol |
| **2068** | Hypertext Transfer Protocol -- HTTP/1.1 |
| **2069** | An Extension to HTTP: Digest Access Authentication |