Setting Up the Domino GO Secure Server

Dr. Virginia R. Hetrick
Technologist

Bellnet: 310.471.1766
Internet: drjuice@gte.net
http://home1.gte.net

Southern California OS/2 User Group
1998 Open House

The major issues required to setup the IBM Internet Connection Server are described in this paper along with some of the "gotchas" we discovered on the way to getting our server up and running.

Secure or Not Secure

The first decision to make is whether the server is to be a secure server or not. The major issue is whether you want to operate using information that most people would not like to have floating around the Internet unencrypted. Among the kinds of information that fall into this category are credit card numbers, Social Security numbers, and medical and financial information.

If, in the long-term, you want to operate a secure server, it is possible to run the Internet Connection Secure Server (ICSS) as an insecure server at first and then add the security features. This saves buying two packages of the software where one would suffice. This is exactly the course we followed at The Institute of Archæology. We ran an insecure server for a while then switched on the security features.

It is also possible to use the server as a caching proxy server if the server is inside a firewall. This minimizes the amount of traffic which must come across the firewall.

If you have a firewall and use the ICSS as a caching proxy, you need to make that decision before you begin setting up the server.

The details for acquiring and managing certificates and keys are better left to another presentation. The major points related to these issues are identified below.

Pre-requisites (required and optional)

We recommend running the Domino GO over OS/2 Warp 4. Three APARs should be installed as well, IN12289, IN15782, and IN16223. In our particular installation, we have also installed Fixpak 5 for Warp 4 on the server.

Version 1.1 is running on our system. Because we backup our computers using ZIP drives, we always add a ZIP drive device to our config.sys file. Our personal preference is to use the OAD driver because the server need not be rebooted when the ZIP drive is attached momentarily to backup this particular system. The one disadvantage of this is that the driver is named os2.sys and must live in the root directory of the c: drive which presents a little confusion the first few times it shows up in a list of os2*.* files! Because ZIP drives cannot yet handle HPFS file systems, it is important to use PKZIP2 to zip the directories into 8.3 format. Of course, if the earlier advice to have an 8.3 name format for files was followed, they can be backed up directly to the ZIP drive.

In any case, the software must be installed on a high-performance file system (HPFS). It will not install properly on a FAT file system. In our case, both disk drives are HPFSs. And, even though the data reside on an HPFS file system, we use FAT file-naming conventions because many people who provide pages to the server are running on DOS or Windows 3.11. Even people using Macintoshes to build their pages are instructed to use FAT file-naming conventions so they can check out their page structures and be certain they work before submitting them for publication. It is handy but not required to have a CD-ROM drive available. We have a BackPack CD-ROM drive which we automatically include in all OS/2 configurations with the /nonstop option specified so that it is not necessary to have a <CR> entered during the boot process.

Sizing the server is an issue for another entire presentation. However, we have found that for a "small" operation, using a 75mHz Pentium with 32 MB of memory and 1.5 GB of harddrive space has sufficed thus far. The data for the entire site, to this point, takes only about 10MB of the d: drive.

We averaged 100 hits per day from distinct individual users to the archæological content areas, mostly during business hours, since we put the server online in late winter 1997. Prior to our registering the site with Yahoo, we averaged about 30 individual user hits per day. The server is presently being indexed by about 25 webots each night. The server is connected to our network cabinet, where the hubs are located, by 10BaseT Ethernet and thence to the campus backbone through a transceiver to a T-1 fiber optic cable.

If you will be running a secure server immediately, you will need to decide whether to self-certify or to have a trusted-third-party certification. The former is useful mainly for Intranet sites while the latter is useful for Internet sites because nearly all third-party certificate providers are already included in the certificate lists for both Netscape and Internet Explorer. If you decide to self-certify, you will need to ensure that everyone accessing your secure site adds your certificate to their browser's certificate list.

The certificates, usually abbreviated CA for Certifying Authority, are available at four levels, curiously enough named levels 1 through 4. In general, most providers are now beginning to conform to OSI X.509 and FIPS 140-1.

In the ICSS manual, only Verisign is listed as a certificate provider. Verisign provides a number of different certificate classes for servers as well as individual digital identities. However, additional providers are available as well. The exact mechanisms for obtaining certificates differ slightly among these vendors, but precise details for each can be obtained from the Web sites listed below. Obtaining the certificate can be a work in progress while setting up the remaining details of the server. Some certificate providers are shown in this list:

Certifying Authority	Web site for further information
GTE CyberTrust	http://www.cybertrust.com
BBN Certificate Services	http://www.bbn.com/products/security/skcms.htm
The US Postal Service	http://www.usps.gov/
AT&T Directory Services	http://www.att.com/
MCI Mall	http://www.mci.com/
Thawte Consulting	http://www.thawte.com/cgi-bin/server/step1.sioux
Note: Thawte also has a good FAQ on the issues of certificates and encryption at: http://www.thawte.com/faq	
CommerceNet	http://www.commerce.net/
KEYWITNESS (Canada only)	http://www.keywitness.ca/
Canada Post Corporation (Canada only)	http://www.post.ca/

Configuring Your File System

It is important to understand the file structure you will be using for your server. For example, in our system, the operating system and software live on the c: drive and the data reside on the d: drive. Unfortunately, to date, no site management tools seem to be available to manage ICS sites. Consequently, when we designed The Institute's site, we took some care to keep things manageable from a file structure perspective. As it turns out, this also means that we have a site which loads fast, regardless of the networkd distance a particular user may be from our location.

In general, using the file structure supplied by the server is probably the best idea. By default, the installation provides a directory structure as below off the root:

www
admin
admin-bin
bin
cgi-bin
docs
html
icons
logs

We then added wiggle for our user directory, as we specified in the administration setup process described below.

In our setup, the entire setup is on the d: drive because it appears not to be possible to put the executables on one drive and the data on another to conform to our customary practice.

However, once installation directory structure is set, you will need to provide a structure for your data, such as .html files, that makes sense to the people who will be maintaining the server. The specific details of logical and visual Web site design are addressed in the paper, ***Designing a Web Site: Logical and Visual Organization***, which is found on our Web site at:

http://www.ioa.ucla.edu/~hetrick/logvis.htm

Here it will suffice to address two of the issues addressed in that longer paper.

First, it is helpful to have the file system's structure reflect the structure of your organization. This also helps your internal users. By using directives judiciously, it is possible to impose a different structure for external users. For example, at http://www.amgen.com the items available to an external user do not, by any means, reflect the internal file structure of the site. The pages available to external users reflect their interests in communicating with Amgen and not the company's need for internal communication.

On our site, the directory structure is reflected in the structure of our initial image map, which can be seen at

http://www.ioa.ucla.edu

Each of the items in the larger type represents a primary directory under the /www/html directory on the site. Each item in smaller type is a subdirectory under its respective primary directory. In general, so far, this has resulted in each of the directories we work with on a day-to-day basis having less than 60 files.

Next, will you have "personal" pages for people in your company? Not all organizations allow these kinds of pages so you may need to have a policy directive.

In general, colleges and universities as well as small companies tend to have these; medium and large enterprises tend not to have them.

The policy for our site is three-pronged. First, each person who chooses to have a personal area on our site must have a front page which contains the details of their research or academic work for The Institute. By convention, the file identifier for this file is index.htm, because we chose that as our welcome page file identifier during configuration. (Other alternatives are Welcome.htm and welcome.htm. The name,

frntpage.htm, is reserved for administering the site.) Second, any additional materials are the responsibility of the author, except for students, and not the responsibility of The Regents of the University of California, our official legal beagles. Third, any student wishing to have a personal area on our site must have the content approved by her/his supervising faculty member. This is largely a matter of (a) archæological correctness and (b) the Web dudette s lack of interest in joining the thought police.

If personal areas are allowed, directories for those need to be directly accessible off the root directory containing the data, if they follow the convention of beginning with a tilde, ~hetrick, for example. One way to accomplish this, without having the unpleasantness of many ~ directories off the root, is to set up directives in the configuration file to redirect requests to an alternative source (details of using directives are described later in this paper). Our personal preference is to put each person's materials in a separate directory so that updates are not too confusing. All of these separate directories are subdirectories of the wiggle directory off the root of the d: drive. Then, using directives, each person's individual directory is accessible at the first level of the Web site.

Installing the Software

Installing the software is an extremely simple process. The software arrives on a CD-ROM but four diskettes are provided as well. The main difference is that the CD-ROM saves having to stuff in the diskettes into the drive so that the installation time is reduced to about two minutes versus five minutes for the diskettes! All that is required is to pop an OS/2 command prompt, point it to the drive where the installation materials can be found, and enter install at the prompt.

You will need to supply the location to install the software, if you do not want to use the default. You will also need to supply an administrator id and password if you do not wish to use the defaults provided (we STRONGLY recommend NOT using the defaults for obvious reasons).

Until you are ready to turn the server loose to the world, we recommend putting the server online for testing only and removing its network connection (or shutting down the server software) at night unless (a) it is behind a firewall or (b) only Intranet connections exist. In this way, external webots are unlikely to find your server until it is completely prepared and you actually want it registered with the various search engines.

Configuring Your Server

Configuration is performed by accessing the server's administrative functions at http://www.subdomain.name/frntpage.htm which will require you to access the administrator id using the password you decided on at installation time.

Directives

A number of different options are available to specify the way in which the server handles requests. Several of them are important to understand to implement your server with only minimal difficulty while others are important for the security of your server. Only a few of the possible directives are covered in this paper.

NOTE: The order in which the directives are issued is extremely important because certain directives prevent other directives from being used as the Webmaster intends.

Only one space is required between character strings; we have separated the strings for clarity in the examples below.

In the absence of a directive, the server expects to find directories beginning with a tilde (~) directly off the root of the data drive, the d: drive in our case. It is probably not desirable to have large numbers of directories directly at the root because this would slow the server's performance. So, we created the wiggle directory at the root and all of the directories for personal pages are in the wiggle directory. The directive to specify this is the HomeDir directive and, for our site, takes the form:

HomeDir d:/wiggle

Why is this necessary, you might ask. If you do not use a directive and have a reference to a page such as mine on the front of this document, http://www.ioa.ucla.edu/~hetrick, the server will expect to find my directory, hetrick, directly off the root level of the drive where the "welcome" page for the entire site lives in the /www/html directory. This is a throwback to the UNIX origins of the Web where the user directory contains each individiual's files, including their own Web pages. Note that, when the tilde (~) is placed in the specification of a URL, the directory to be accessed will be in the HomeDir without the tilde.

The next directive we will look at is the PASS directive which redirects a request to the server to some other location on the same server. The general structure is:

Pass request-template [file-path]

By default, all requests are passed. Wildcards are allowed except for tildes (~) which must be explicitly matched. So, for my particular directory, the directive would read

Pass ~hetrick d:/wiggle/hetrick/

Next is Redirect which redirects a request to the server to some other server. The general structure is:

Redirect request-template full-URL-and-path

This directive is also useful if you should, for some reason, decide to change the structure of your server or to split the server's functions among several physically separate computers, i.e., not SMP or other multiprocessor machines. Failing to use the Redirect directive will result in the appearance of the dreaded HTTP 1.0 404 message, Object not found on server. For example, if my pages were to be moved to a server on mycomputer, the directive would be:

Redirect ~hetrick http://mycomputer.ioa.ucla.edu/~hetrick/

The next directive is Disable which provides some degree of protection from users' attempting to delete objects on your server. The general structure is:

Disable method-request-template

By default CHECKIN, CHECKOUT, DELETE, and PUT are Disabled.

Enable is the next directive. It specifies which HTTP methods are allowed on your server. The general structure is:

Enable method-request-template

By default, GET, HEAD, and POST are Enabled.

The next directive is DirAccess which controls whether or not users can see the contents of the server's various directories. By default, it is set On. We have set this to Off as a security measure since we provide private directories, outside the scope of the server, for FTPing files.

If you will be using the server as a caching proxy, it is common to use a different port than the default (Port 80) for the server to listen for requests. The most frequently used alternatives for caching proxies are 8080 and 8008. If other than port 80 is used, the client must specify the port in the URL, for example:

http://www.myproxy.com:8080

Either two groups or three groups of log files are created and need to be cleaned up from time to time, particularly if you are running the server on a computer that performs other tasks, such as a file or print server. All logs are daily logs, being started afresh at midnight each day. By default, they are created in the directory, /www/logs, which may be changed at installation. The time stamps in log files are local time stamps by default. All logs have a stem indicating their type and an extension representing the date. The three types of logs are:

- httpd-log Logs access to the server, on by default
- error-log Logs internal server errors, on by default
- cache-log Logs cache access, off by default

In Summary