Notes.net

Iris Today

Overview of Notes/Domino security

by Susan Bryant
and Christie Williams
(with Katherine Spanbauer)

Level: All
Works with: Domino 5.0
Updated: 09/04/2001

Truth be told, if most network administrators had their way, networks would operate in a vacuum, with the access provided only to the administrators themselves. Instead, administrators must simultaneously allow and control access to the servers and the data on those servers for users within the organization as well as users of the Web.

It's no surprise that providing access to data and controlling that same access is one of the greatest challenges facing corporations. We've all heard stories—and seen a myriad of advertising—about the tragedies incurred when networks are compromised. The consequences can range from mildly annoying to severely damaging, costing millions of dollars (or yen, or francs, or pounds) and resulting in loss of service—and loss of employment for the people who should have been securing the network. The challenge for network administrators, who are most often charged with protecting the corporate data, is to provide balance for this seeming dichotomy.

Fortunately, a Domino network provides a full complement of security features, any or all of which can be used to protect your servers and Notes workstations and the data contained on them. This article provides an overview of the security features available in a Notes/Domino environment and how they let you allow seamless access to those users who should have access and deny access to those who shouldn't.

The Domino server: Provider and protector

The Domino server software does just what the name implies: it serves up data either to Notes clients, Web clients, or both. If we, as administrators, do our job correctly, it will serve this data only to those authorized to have it.

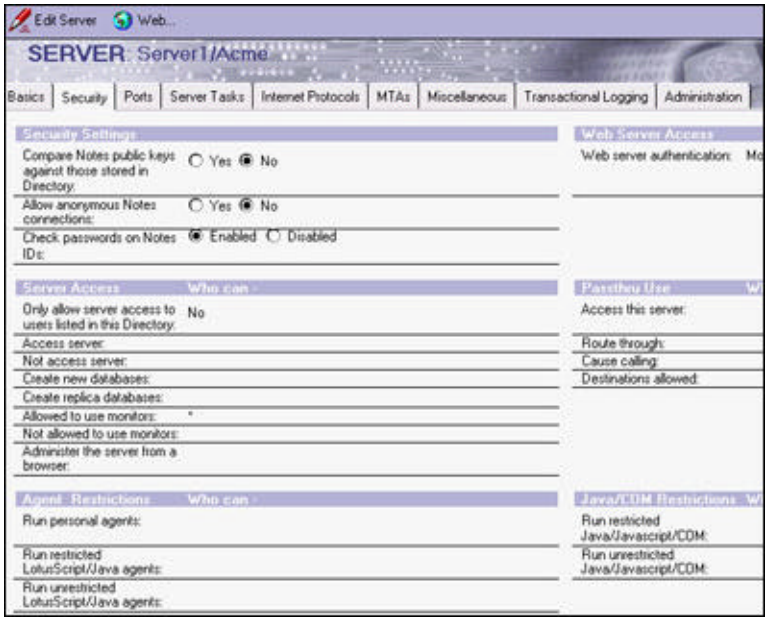
It's important to note a few things right off the bat:

- Many of the security feature default settings when a Domino server is built allow very open access to your servers and their applications, so it's important to become familiar with all aspects of Notes/Domino security and to develop an effective security plan.
- Domino verifies and authenticates Notes clients differently than it does Web clients. If your server is providing data to both types of clients, it's important to know how each is authenticated and to know which security features work with which client.

The first line of defense: The Server document

Although the true first line of defense is assuring that the operating system is secure, for this article and because Domino can be installed on most operating system platforms, we'll assume that you've already secured your respective operating system.

When a client makes a request to the Domino server, the first check that the server makes is to verify that the client has access to the server. Each server has its own security settings as defined in the Security section of the Server document in the Domino Directory.



Appropriate entries in these fields (user names, group names, as well as some wildcard entries) can go a long way to securing the server. These settings control who can access your server, who's denied access to your server, how the client is authenticated, who can run various types of agents, as well as various security features. It's important to become familiar with these settings (many of which are discussed in this article) as this is truly your first line of defense for the Domino server.

Authentication
Server document fields come into play during client authentication. Because the Notes client and a browser client have different capabilities, authentication for each is different. Let's look at Notes authentication first.

Notes client authentication
The Notes user ID file contains all the information needed for a user to identify themselves to the Domino server. This includes the user name, password, and appropriate certificates for the organization. Before connecting to the server, the user must enter their password correctly. (See the **The Notes user ID and password** section below for more about user IDs and passwords.) Then, to establish a connection with the Domino server, all certificates stored in the ID are sent to the server. The server validates the certificates stored in the ID with the corresponding certificates in the Domino Directory and assures that the client is valid—or access is denied.

After the client is validated (that is, the certificates are trusted), the authentication process proves that users really are who they claim to be by establishing a challenge/response dialog between the workstation and server. The same validation and authentication happens between two servers when the servers establish connections for replication or mail routing. Simply put, the server and workstation (or two servers), are exchanging encrypted random numbers, decrypting the numbers using the public keys that are stored in the IDs and the Domino Directory, and sending them back.

The user name is compared with Person documents, Group documents, and Server document access fields; and if the user name doesn't pass these tests, access to the server is denied. If all is well—the client can be validated and authenticated and that valid and authenticated user is not

denied access to the server by means of various fields in the Server document—then the client is good to go.

Once a session is established, the user will not be prompted to identify themselves again during the session unless the ID is locked. The ID can be locked manually (and the session dropped) by pressing F5, or the user ID can be configured on the User Preferences dialog box to be locked (and the session dropped) after any number of minutes of inactivity. The password must be entered again to reactivate the session. Locking the ID prevents someone else from sitting at your machine and using your identity if you leave the machine unattended.

Special Notes client authentication scenarios

There are variations on this Notes client authentication process, which you can implement using fields in the Server document.

- Anonymous access
  As described above, the authentication process checks certificates in the Notes ID with certificates in the Domino Directory. However, you may want users and servers outside your organization to have access to databases on your server without obtaining a certificate. For instance, your may have a discussion database for your customers; it would be unwieldy to cross-certify every user from every customer organization, and since the information in the database is not sensitive, it would be unnecessary as well. By allowing Anonymous access using the "Allow anonymous Notes connections" field in the Server document, this non-authenticated access is possible. (Access restrictions to the various databases on the server are then controlled with each database's Access Control List, which is discussed in the **The database Access Control List** section below.) Non-authenticated users have an access level of Anonymous, but if an Anonymous entry does not exist, then the access rights assigned to Default apply to these users.
- Password checking
  To ensure that users change their passwords on a regular basis, you can turn on password checking using the "Check passwords on Notes IDs" field in the Server document. Then users must change their password within the chosen time interval or they are locked out of the server during authentication and must rely on an administrator to reinstate them. For details about administering password checking see the *Iris Today* article "**Notes from Support: Password checking**."

Web client authentication

Because the Web client has no Notes ID, verification of the client happens in a different way. Instead of authenticating when the browser client connects with the server (as happens with a Notes client), authentication occurs when the browser attempts to access a server with Anonymous access disabled (via the Ports/Internet Ports tab of the Server document) or a database on the server that does not allow Anonymous access or whose Default access is set to No access. (If the database's ACL allows Anonymous access or has its Default access set to Reader or above, then no authentication occurs; anyone can access the database.)

If the server or database has been secured, however, then the client is prompted to enter a name and a password when they try to access that data. This information is verified by comparing the supplied name and password against the Person document in the Domino Directory and the Internet password field in that document. If the name and password provided don't match those stored in the Person document, access is denied.

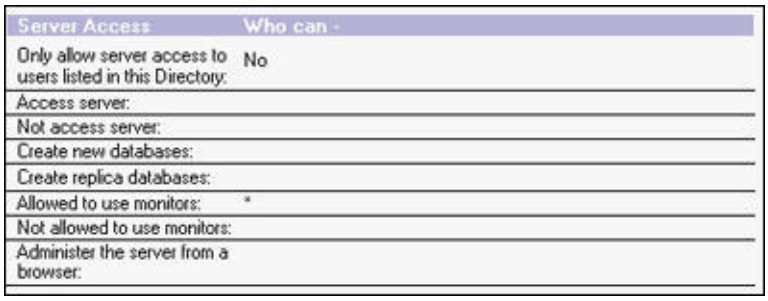
This name and password information is stored in a packet header sent to the server for verification. Obviously, this information can be intercepted by anyone with a sniffer or trace tool. The name and password information is also cached on the browser (so that you are not repeatedly prompted for

this information). Users should be aware that because of this caching, if they leave the machine unattended during a session, anyone using that machine can use their identity. If they share a machine or use a public machine, they should be sure to end the session by closing the browser before leaving the computer.

So, how do you protect sensitive information and still open your servers and application to the Web? By using Web standards, or course.

The Domino server supports the Internet X.509 standard for public key certificates. These certificates are used for secure Internet transport for SSL and S/MIME (explained in the **Secure messaging** section below). Additionally, the Notes client support for Internet protocols allows you to store X.509 certificates in your ID file for use in securing transactions with other Web servers over SSL, and to sign and encrypt mail to Internet mail users using S/MIME.

Server access controls
During authentication of the Notes client, Domino checks server access fields in the Server document to see if the user (or server) is allowed access. For example, setting "Only allow server access to users listed in this Directory" to yes, limits access to only those people, groups, and servers who are in that Domino Directory. A setting of no, opens access to people, groups, or servers in other Domino Directories. With the server access fields, you can specify people, groups, or servers that can access or not access the server, create new databases or new replicas of databases, and so on. (These settings do not apply to client access via Internet protocols—a Person document in a Domino or LDAP directory, configured via Directory Assistance, is required to in order to authenticate Internet users.)



Server Access	Who can -
Only allow server access to users listed in this Directory:	No
Access server:	
Not access server:	
Create new databases:	
Create replica databases:	
Allowed to use monitors:	*
Not allowed to use monitors:	
Administer the server from a browser:	

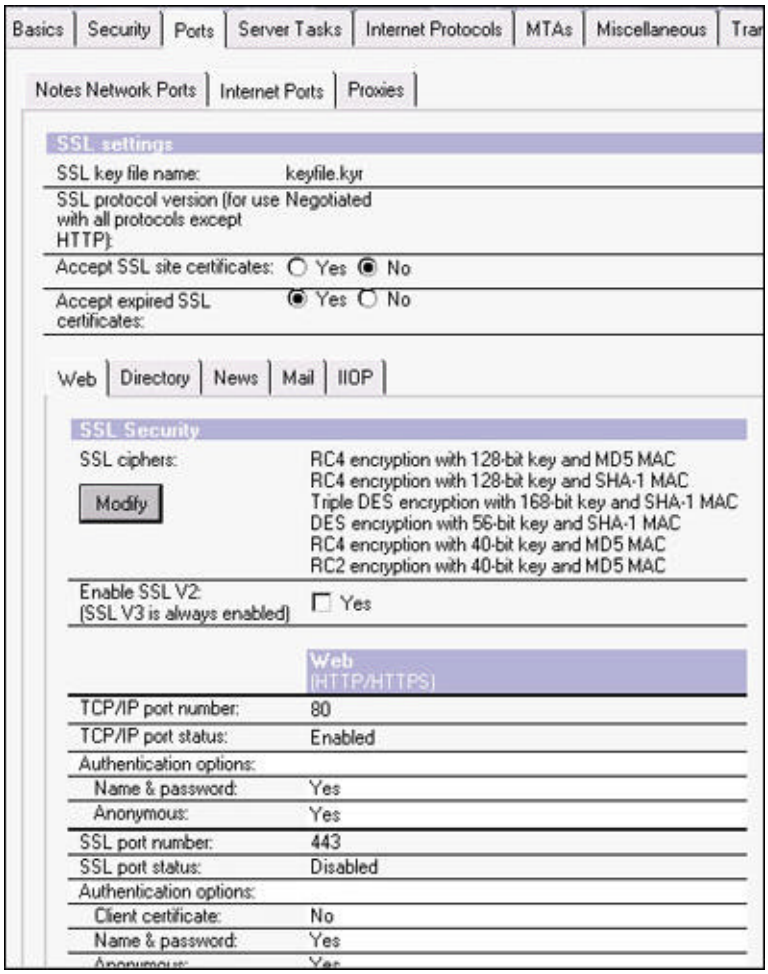
Note that the Server document has similar deny/allow access settings for passthru servers.

Securing server ports
All of the server's network ports that users use to access it can be secured using various security standards. Additionally, Notes clients can encrypt network traffic for individual ports (see the **Client port encryption** section below).

In addition to overall server access as outlined above, individual LAN ports on the Domino server can be configured to encrypt network traffic over the Notes protocol using the Server tool, Setup Ports, in the Domino Administrator client.

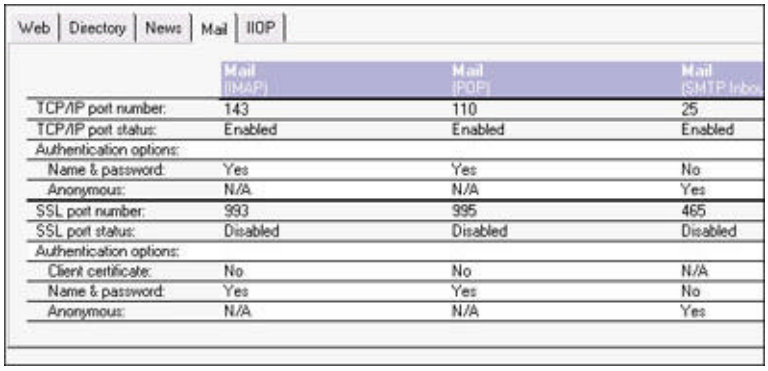
Because data that travels through the Web can be intercepted, it is often necessary to secure confidential data or transactions. The Domino server provides support for SSL port encryption for all Internet ports to further protect the requests made to the server from the Web. SSL (Secure Sockets Layer) is a security protocol that provides privacy and authentication for Internet traffic. These SSL settings are also defined in the

Server document.



SSL authentication is similar to Notes authentication. The SSL client presents a certificate to a Domino server. Then the server uses that certificate to authenticate the client, and vice versa. Unlike Notes and Domino, SSL does not require that both the server and client authenticate each other. If SSL is enabled, Domino requires that the client authenticate the server; however, it is optional for the server to authenticate the client. SSL uses an Internet certificate in X.509 format, which is an industry standard format that most applications support, including Domino. The Domino server can use a server certificate signed by a public, or external, Certificate Authority (Verisign, GTE, Thawte, and so on) or can act as its own Certificate Authority.

The Domino server can act as a Web virtual host or virtual server. Virtual servers in R5 can each have their own SSL configuration. The Domino server can also serve mail to various standard mail clients, serve as a network news server, and serve as an LDAP server. Each of the ports used to perform these tasks can be secured.



Web | Directory | News | Mail | IIOP

	Mail (IMAP)	Mail (POP)	Mail (SMTP Inbo
TCP/IP port number:	143	110	25
TCP/IP port status:	Enabled	Enabled	Enabled
Authentication options:			
Name & password:	Yes	Yes	No
Anonymous:	N/A	N/A	Yes
SSL port number:	993	995	465
SSL port status:	Disabled	Disabled	Disabled
Authentication options:			
Client certificate:	No	No	N/A
Name & password:	Yes	Yes	No
Anonymous:	N/A	N/A	Yes

Additional server security considerations
Here are a few other things to consider regarding server security.

Controlling administration of the server
Performing administrative tasks on the Domino server is made easier through the use of the remote server console. Administrators can change settings, observe situations, or reboot the Domino server from nearly any Windows NT machine using the Domino Administrator client or a Web browser. This is an awesome responsibility, and the security to this privilege should be strictly controlled. Domino provides separate settings in the Server document to control who can perform remote administration either from the Domino Administrator client or a Web browser.

Monitoring the server for attacks
The Domino server log (log.nsf) provides a play-by-play account of what is happening on the server. A quick search of this database can tell you if there are security or access concerns. The server console can also give real-time information about what's happening with the server, but few administrators have time to sit and watch the chatter on the server console (unless you're paid by the hour and even then it's still a tedious, boring task.)

It's more effective, not to mention efficient, to use the Collect and Event (or Event Dispatcher) server tasks to monitor attempted but failed access to a server or database and notify you of a problem. You can configure the server to notify the administrator by collecting the information into a separate database, mail a notice to the administrator (which, of course, can be configured to page the appropriate technician), send a trap to an SNTP trap on your network, log to a UNIX system log or the NT Event Viewer, or relay to another server.

Disabling the OpenServer URL command
The special URL, http://myserver/?OpenServer, generates a page containing active links to all the databases on the server. Access to this list of all databases is convenient and useful for administrators or application developers working on a Web site, but it is rarely, if ever, appropriate for the general public. However, the setting in the Server document that controls the OpenServer URL command, "Allow HTTP clients to browse databases," is all-or-nothing. There is no way to limit which users can see the database list or which databases appear on the list.

To ensure site security, you may want to disable the OpenServer URL command entirely by setting "Allow HTTP clients to browse databases" to no. (See the *Iris Today* article, "**An alternative to the OpenServer URL command**," for information about an agent that lets you specify who has access to the list of databases on your server.)

Securing the client

Securing the client is as important as securing the server. You've seen how both the Notes client and the Web client authenticate with the server. In addition, there are additional client-related security features.

The Notes user ID and password
The first line of defense for the Notes client is, of course, the user ID and password. When a Notes client is registered, the user name and appropriate certificates for the organization are stored in the Notes user ID file. Corresponding certificates for each user are also stored in the Domino Directory. When the IDs are created during registration, you can define a password as well as the complexity, or quality, of the password. Passwords with a higher password quality scale rating are more secure than those with a lower rating. (See the *Iris Today* article, "**Understanding password quality**," for more information.)

Protecting the ID with a password means that no one can use the ID to access a Domino server without knowing the password. When a Notes client is started, the user is prompted for the password to this ID. Without it, beginning with Notes R5, the Notes client cannot be started. In all releases of the Notes client, without the password to the ID, the client cannot access the Domino server.

Secure messaging
Notes provides the ability to sign and encrypt e-mail messages to other Notes users using Notes protocols and certificates, as well as to other Internet mail users using the S/MIME protocol and X.509v3 certificates. Both of these methods are based on public key technology developed by RSA.

If you are sending e-mail within your secure network or intranet, signing and encrypting mail may not always be necessary. But if your message is traveling the Internet, you cannot guarantee that the necessary security precautions have been taken at every hop along the route. Therefore, the only sure way to ensure the confidentiality, authentication, and integrity of your message is to make sure it is encrypted and/or signed.

When sending an e-mail message, you can select to sign or encrypt the message from Delivery Options in the mail file. If sending to another Notes user, native Notes signing and encryption will be used; if sending to an Internet user, S/MIME will be used. You can also select to sign or encrypt all sent messages and encrypted saved messages by default in the User Preferences dialog box. In addition, you can encrypt incoming mail using a field in the Person document; when the router delivers mail to your server, the messages will be encrypted using your Notes public key.

To send an encrypted message, you need to obtain the recipient's public key and encrypt the message using this key. In many cases, these keys are stored in the Domino Directory or an LDAP directory and can be retrieved from there. However, when sending mail outside your organization, you may not have access to a directory that has the keys. In this case, an easy way to do this is for users to exchange S/MIME signed mail and copy the sender's key into their personal address book (choose Tools - Add Sender to Address Book). To exchange your Notes public key with users outside your domain, use the Mail Public Key button on the More Options panel of the User ID dialog box (choose File - Tools - User ID). Notes will check your personal address book for certificates first, then the Domino Directory on your home server, and finally any directories configured via Directory Assistance. When the recipient opens the message, it is decrypted using their private key. Notes stores the private keys in the user ID file.

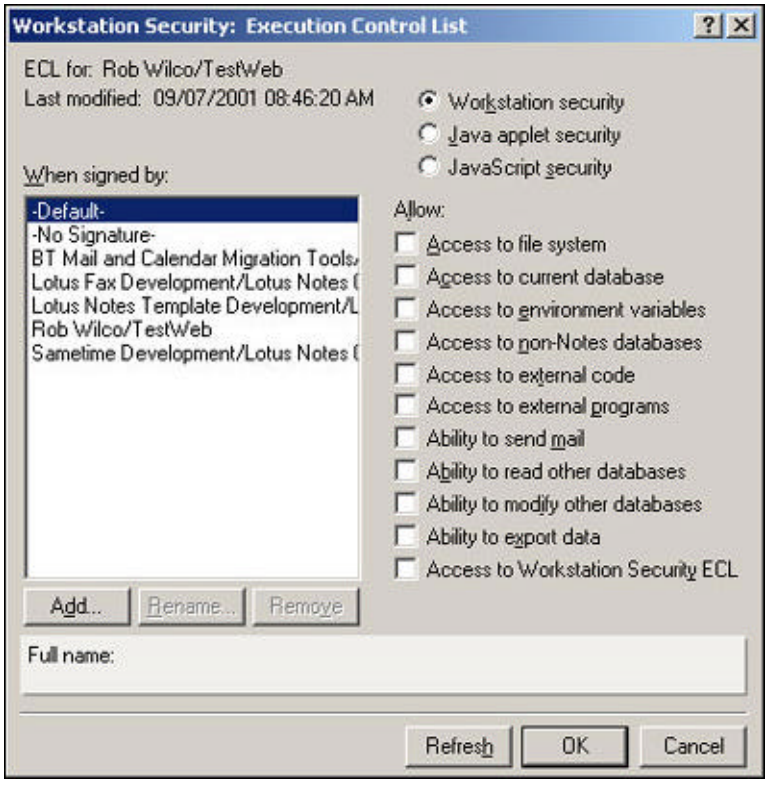
Notes also provides message signing by using digital signatures. Messages are signed with the sender's private key. The signed message is sent with a certificate that vouches for the authenticity of the sender’s public key. The

recipient can verify the signature with the sender's public key as long as the certificate was issued by a trusted Certificate Authority.

To be able to send signed or encrypted mail using S/MIME, whether you're using the Notes client or another S/MIME compliant mail client, you need to obtain an X.509 certificate from a trusted Certificate Authority.

Execution Control Lists
ECLs (Execution Control Lists) impose security restrictions on Notes formulas and external programs such as JavaScript and Java. The database manager and system administrator can protect users' workstations and databases from unauthorized access by creating an ECL for each workstation and by signing templates and databases. ECLs determine the tasks that a procedure embedded in a database can perform in the database and on the workstation according to who signed the procedure. For example, you may give limited execution access to your Domino system administrator, but allow no execution access to unsigned scripts or formulas.

Individual users can edit their ECL to suit their needs for security (unless the administrator has disabled that capability). The user's ECL options appear in the Workstation Security: Execution Control List dialog box, which is accessible from the Basics panel of the User Preferences dialog box.



Additionally, administrators can (and should) create an Administration ECL that is stored in the Domino Directory and gets copied to the user's workstation when the system administrator registers a new user. This feature facilitates standardizing workstation security throughout the corporation.

A Notes workstation ECL can limit the following:
- Access to the file system
- Access to current database

- Access to environment variables
- Access to non-Notes databases
- Access to external code
- Access to external programs
- Ability to send mail
- Ability to read other databases
- Ability to modify other databases
- Ability to export data
- Access to Workstation Security ECL

When a Java applet runs within Notes, certain security restrictions are imposed on that applet. This security model protects against malicious code by determining what operations an applet can perform and what system resources it can access. These restrictions can be customized on a per-signature basis. The Java ECL can limit the following:

- Access to file system
- Access to Notes Java classes
- Access to network addresses
- Printing
- Access to system properties
- Dialog and clipboard access
- Process-level access

The JavaScript ECL options control security for JavaScript that runs within the Notes client, either on a Notes form or on a Web page rendered by the Notes browser. These options do not control JavaScript that other browsers (such as Microsoft Internet Explorer) run, even when the JavaScript is embedded within the Notes client.

You can control the security for the read and write options independently for three different classes of Window objects:

- Source window
- Other window from same host
- Other window from different host

There are also two additional ECL options that control whether JavaScript executing in the Notes client is authorized to open a new Web page or Notes document.

For more details about ECLs, see the *Iris Today* article, "**Staying alert with Execution Control Lists**."

Client port encryption

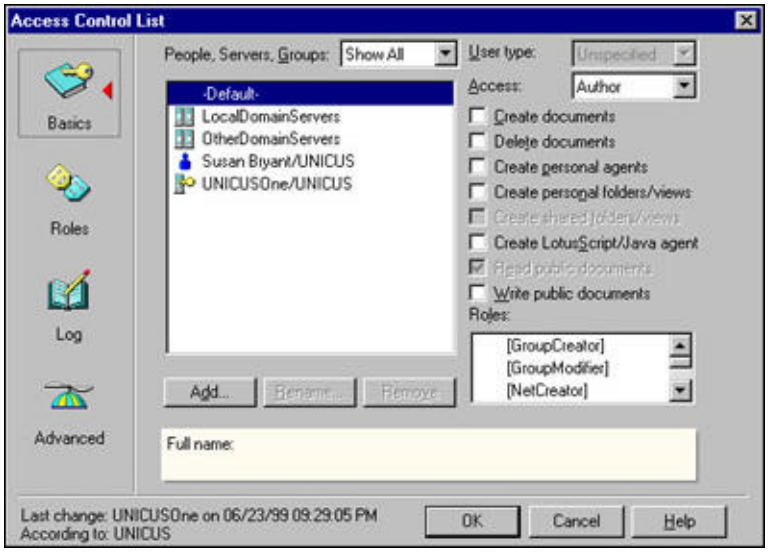
Users can also choose to encrypt network data using the Ports panel of the User Preferences dialog box. This protects data in transit between the client and server and is useful if the server does not have port encryption enabled. This setting only needs to be enabled on one side for the entire transaction to be encrypted. See the **Securing server ports** section above for more information about port encryption.

Securing the application

Beyond controlling access to the Domino server, the administrator has to work with the application designer to decide what data the users should have access to. Domino allows very granular security control to the data in an application, whether it's stored in a Notes/Domino database or contained within individual HTML files.

The database Access Control List

Once access is allowed to the server, the server checks access to the data itself. If the data being requested is contained in a Notes/Domino database, the first access point to the database is the Access Control List.



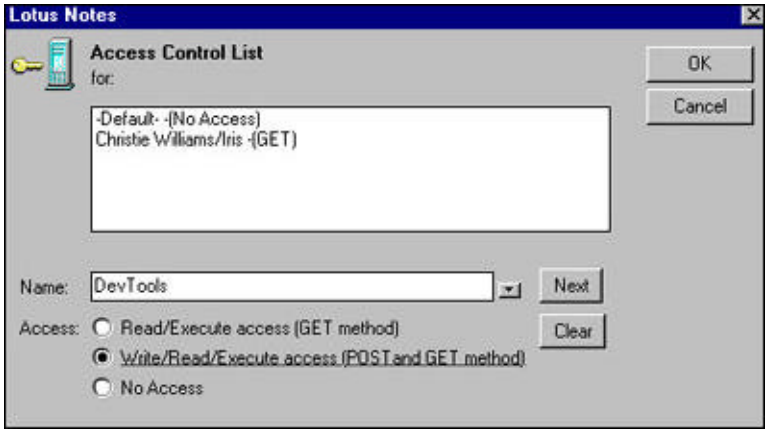
The Access Control List controls access to each Domino database application. A client's access level can be different for each application or the administrator can more universally control access by creating groups of users and assigning access to the group. Much has been written about controlling application access by properly configuring the Access Control List. Please see the *Iris Today* article, "**The ABCs of using the ACL**," as well as the **Domino R5 Administration Help**.

It's important to have a complete understanding of how to use the Access Control List, since it is the first access point to a Notes/Domino database.

Access control for the file system

Beyond securing your operating system, if your Domino server is open to the Web, you may want or need to configure access to certain files on the server that are not Notes databases. This is especially true if the Domino HTTP server is serving HTML pages that are sitting out on the disk. While you do not have the granular control of access to file system files that you have to a Notes/Domino database, Domino provides some general control. Domino supports standard HTTP security around the GET and POST methods.

In R5, you configure this control from the Server document in the Domino Directory by clicking the Web action button and choosing Create File Protection to create a File Protection document for that server. On the Access Control tab of this document, you can set or modify the access control list for files on the server.



Security-related database properties

Local database encryption

When your laptop is stolen in an airport, you may be upset that you just lost your DVD player, but it's very likely that the most valuable thing on that laptop is not the hardware or the software, but the data. Notes databases stored locally are, by default, easily opened and accessed because they are not protected by a server and all its settings. Databases containing sensitive information can be encrypted with a user's ID so that it can only be accessed when one knows the password to that ID.

Using the Encryption settings in the Database properties box, you can encrypt a local database with any specified user or server ID. This feature was primarily designed as a workstation feature, since physical security of workstations, especially laptops, cannot be as tightly controlled as that of a server. However, databases on the server can also be locally encrypted using the server ID, although this is not a common practice. In order to be effective, the ID used to secure the database must be protected with a password.

Requiring an SSL connection

For any database, you can require an SSL connection for Web access, which enforces the use of SSL for that particular database. This is useful if SSL is not required for all databases on the server as it allows you to provide SSL security to any individual database, for example, one with sensitive data.

Controlling access to data using design elements

Domino provides a rich palette of security features within a database to secure the data it contains. Data is processed, added to, and accessed from a Notes/Domino database using a variety of design elements—forms, views, agents, fields, and so on. Most design elements have security features that allow the administrator or the database designer to control who has access to using the design element and the data created from the design element. For more information about these design elements, see **Domino R5 Designer Help**.

View properties

Security options in the View properties box allow an administrator or designer to define who—individually, as a group, or in a role—has access to the view. However, Notes client users may be able to create private views to circumvent the view access list, so further measures, such as reader names fields, should be taken to assure controlled access to documents.

Form properties

You can control who can read or create and edit documents based on a

particular form by specifying a list of users, groups, and access roles who need this access, in the design of that form. If a person is not on the list—either individually or as a member of listed group or role—they will not be able to even see the form listed on the Create menu, much less create or edit a document. (This approach helpfully prevents users from receiving frustrating access-denied messages.)

Field properties
Additionally, the designer can restrict the ability to edit a field to only those users who have at least Editor level access to a database in the ACL. This is one of the security options for editable fields on the Advanced tab of the Field properties box.

Author and Reader fields
Author type data fields on a document can control who is allowed to edit a document once it is created if the user has only Author level access to the database in the ACL. The designer can hard-code entries in this type of field, create a formula to determine what entries are placed in this field, or make the field editable and allow the users of the application to determine the list of people who can edit the document. The latter is helpful if the application is used in a collaborative work environment where each project/document may need a unique set of editors and that set will be determined when the document is created.

Reader type data fields are used to determine who can read a document once it is created. Reader fields can refine and override the setting in the ACL. For example, even if someone has Manager level access to the database in the ACL, if there is a Reader field in a document and the manager's name isn't in it, the manager won't be able to read the document. If a user's name is not in a Reader field (and there is at least one entry in that field), the document will not even be visible in any view. This is a very powerful document security feature. Even if a user has the proper ACL access to open and read a database, they may not be able to view or access all of the documents in it.

Controlled Access Sections
Portions of data on a document can be "roped off" in what are called Controlled Access Sections. Defining a list of users who are allowed to edit data within this section prevents anyone else editing the contents of fields in this section, even if they are allowed to edit other fields in the document. They can still view these fields, but they cannot edit them. The section can be hidden, expanded, or collapsed based on the user's current mode (Read, Edit, and so on) and the type of client a user has (a Web browser or Notes). Access lists for the Controlled Access Sections can be hard-coded or controlled programmatically.

Signed documents and fields
Digital signatures can be attached to entire documents or to fields within a document. A digital signature is the electronic equivalent of a handwritten signature—a unique block of text that verifies your identity—that is appended to a message. It can be used to confirm the identity of the sender and the integrity of the message. The block of text is encrypted and decrypted using public and private keys.

Sign-enabled fields on a form allow a digital signature to be attached when a document is saved or mailed. Digital signatures verify that authors are who they say they are and guarantee that the information in the document has not been tampered with. The private key in a user ID file generates the signature. When a user opens the document containing the signed field, Notes verifies the signature by comparing it with the author's public key in the Domino Directory. When a sign-enabled field is placed within a section on a form, the signature appears across the top of the section if the signature can be verified when the document is opened.

Again, because signing a document is done using the user's private key (which is stored in the Notes user ID) and verification of the signature is made by comparing to the user's public key stored in the Domino Directory, signing of fields and documents (including e-mail) is not available for Web clients.

Field encryption
Field data can be encrypted using encryption keys, which are created by one user and distributed either via e-mail (which should also be encrypted) or by exporting, distributing, and then importing the key. The keys are stored in the Notes user ID. When a document that contains encrypted fields is opened for reading or editing, if the necessary keys are present in the user's ID, the fields in the document are decrypted and the data can be viewed. This is completely seamless to the person accessing the document.

Encrypted fields cannot be displayed in a view because they are not decrypted until the document is opened. The encrypted data cannot be viewed from a document's properties in a view, providing true security, unlike hide-when formulas. Encrypted fields cannot be read by Web clients because the necessary decryption/encryption keys are stored in the Notes user ID, which is not used by the Web client. For more details about field encryption, see the *Iris Today* article, "**Using field encryption in applications**."

Using hide-whens
Portions of text and/or data fields on a document can be hidden from certain users by using hide-when formulas. Although hide-whens cannot be classified as a security feature, they may function as "security by obscurity." This is often safe, but it is not secure. Data hidden with hide-when formulas can be seen from the view or document by accessing a document's properties in the Notes client, so it's only "secure" as long as the user is ignorant of this ability. Hiding fields or data also does not protect the data from being modified by agents, actions, or other programmatic access. However, it's a nice feature that allows the designer to selectively display (or hide) data based on a conditions defined in the formula. See the *Iris Today* article, "**Revealing the hidden secrets of 'hide-when'**," for more information about hide-when formulas.

Preventing edit mode
A document can programmatically be prevented from being put in Edit mode as it is being opened in the client even if the user has the appropriate ACL and document access, providing yet another level of security. Programmatic control of Edit mode does not prevent data from being modified by agents or actions, but it does allow yet another way for designers to control how a document and its data are handled.

Notes and Domino: A wealth of security options

Notes and Domino present network administrators with a complete array of security options for assuring that the servers and the data they protect is available to those who need access and protected from those who have no need for it, as well as for data that travels through the Internet. Most of these options are configurable for your particular network. The Notes client can also be secured for local access, and there are innumerable ways to control access to all or some of the data in any database. All together, these features provide a wealth of security options so that you can pick and choose exactly which are right for your situation.

About Susan Bryant
Susan is a Certified Lotus Notes Principal Application Developer, System Administrator, and Instructor. She's been gainfully employed around Lotus

Notes/Domino for over six years. She can be reached at **smbryant@mediaone.net**.

About Katherine Spanbauer
Katherine is the Product Manager for Security, primarily focusing on Notes and Domino. Her current responsibilities include representing customer requirements to development, triaging critical issues, and communicating product features both within Lotus and to customers. Since joining Lotus in 1992, she has held various roles in the Technical Support, Professional Services and Product Management organizations. Katherine is a graduate of the University of Wisconsin, where she earned her Bachelor of Business Administration degree.