



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at <http://www.giac.org/registration/gsec>

Lotus Notes and Domino Security: An Overview of Authentication and Access Control

Craig Barber

Introduction

Lotus Notes was first introduced by Lotus in 1989. Now in its fifth version, Lotus claims that Notes was the first viable groupware product. As a competitor to products such as Microsoft Exchange and Novell GroupWise, Notes is in reality much more than an e-mail package. Notes databases can be used for many types of information management from complex document libraries, transaction records, web based data collection, to simple organization of project deadlines and milestones. The two major components of the Notes system are the Notes Client and the server component called Domino. Notes and Domino together comprise a highly secure, mail-enabled, multi-platform, open, client/server, distributed database management system. Notes has many features that are enabled right out of the box, and its default security features are very good. However, in order to truly be utilized, they must be understood by the administrator.

The scope of the work presented here will concentrate on the security model employed by Notes, and focus specifically on its application of authentication and access control through the Notes Client. Authentication and access control to Domino applications from web browsers is not examined here. It is the intent of this work to help the Domino administrator better understand the process of authenticating to Domino servers and how access to Notes applications is restricted and refined.

Authentication and Access Control

Being a client/server application, in order to gain access to information in a Notes application you must be authenticated by the Domino server. Authentication is the process of proving user identity to the system. Once the user has proven his identity, access to resources can be controlled by the system administrator. Notes/Domino does an excellent job with authentication and access control. Both of these processes are far more in depth than one would expect them to be. The Domino authentication process is very thorough and its access control features allow for very granular control of resources by the administrator.

Authentication: So Just Who Are You?

In any computer information system, there are three types of authentication. Access is granted based on something the user knows, something the user has, or a combination of knowing something and having something. Notes uses the combination of knowledge and possession to authenticate users.

Passwords are the usual piece of knowledge most systems require for authentication. Notes authentication does require a password. However, it also requires the user to have an ID file that is associated with that password. So in

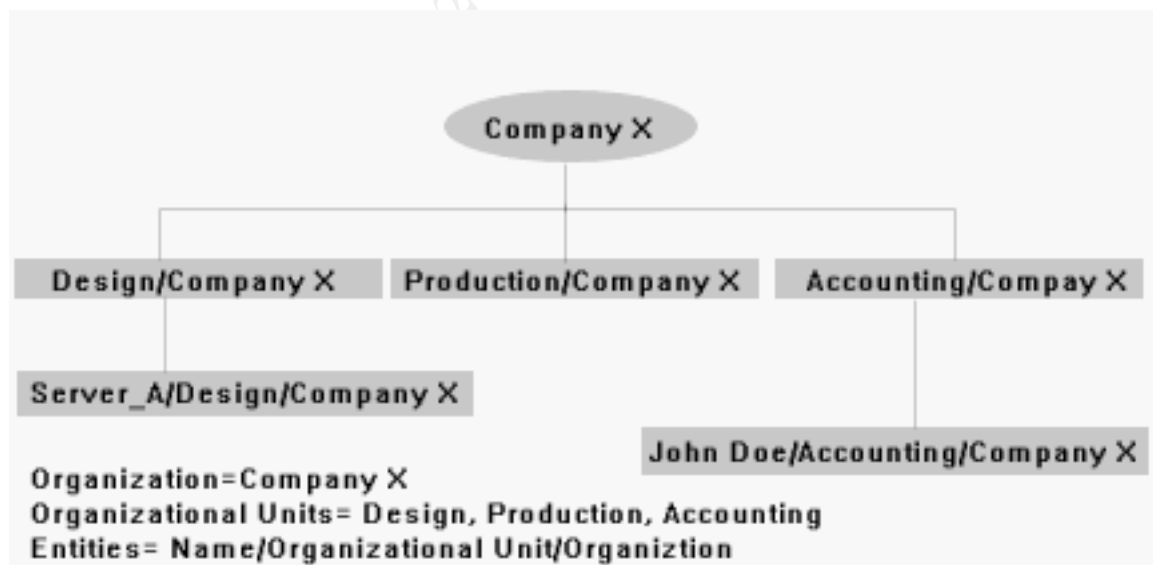
order to gain access to the Notes system, you must have electronic access to the ID file and know the password that goes with it. This means that passwords that are socially engineered from users or discovered by other means are not enough to access the system from any location other than the user's computer.

So, What Is in the ID file?

Explaining the Notes ID file starts the discussion of how Notes/Domino uses certificates and public key encryption for authentication. Notes IDs are binary files that store certificates and encryption keys. Users and servers each get ID files when they are registered into the Domino Directory by the administrator.

The Domino Directory is the administrative foundation of the Notes system. It is a hierarchical, administrative structure that organizes users, groups and servers. In this architecture, entities are grouped into logical organizational units within the directory.

For example, if Company X has different divisions, the directory would set up the Organization as Company X, below which would fall the divisions of Design, Production and Accounting. Within each organizational unit for the divisions would exist users, groups and servers. So, if John Doe works in the accounting division, he would be identified in the Domino Directory as John Doe/Accounting/Company X. A server in the Design division would be identified as Server_A/Design/Company X. (This type of architecture is based on the X.500 standard and is also used by Novell Directory Services and Microsoft Active Directory, among other information systems.) The figure below demonstrates the hierarchy of this example Domino Directory.



When a new user or server is created in the Domino Directory, two RSA key pairs (public and private keys) are generated specifically for that user or server. Then the administrator registers the new entity against another specialized ID file

called the Certifier. This is the process that creates the new certificate for that entity and uses the Certifier's private key to sign the certificate. The signed certificate is then placed in a Notes ID file.

After the registration process, the Notes ID file contains:

- **The user's name and Notes license number**
The user is named in the hierarchical format consistent with the Domino Directory (John Doe/Accounting/Company X).
- **Two public and private key pairs**
These are the RSA key pairs generated when the ID is created in the Notes system.
- **Two certificates for the user**
The certificates contain: The owner's name, the owner's public key, the certifier's name, the Certifier's public key, the certificate expiration date, and a digital signature by the Certifier using the Certifier's private key.
- **A certificate for each ancestor Certifier**
These certificates are for the organizational units above this entity in the directory. So if our user is John Doe/Accounting/Company X, then his ID file also contains certificates for Accounting/Company X and Company X.
- **(Optional) Recovery information for the ID file**
The recovery information can be used by an administrator when a user forgets the password for the ID file or it becomes corrupted in some other manner.

Tell Me More About The Password

Passwords are always targets for security breaches, and a security analysis of authentication for a product must include details on the passwords used by the system. Notes passwords are associated with the specific ID file the user possesses. Which means that another copy of an ID file for the same user could have another password.

In most Domino environments administrators often retain copies of the ID files for all users. Of course they usually know the passwords for these files. This is a wonderful administration tool for resolving situations in which access to a personal Notes database, like an electronic mailbox is ordered by executives in the organization. However this means that the file system where the copies of the ID files are stored must be as secure as possible. We will assume here that Domino administrators who choose to keep copies of ID files will secure them properly.

In regards to the password itself, the Notes password is encrypted in storage and in transmission to the server. The hash used to encrypt the password has come under some fire recently, but Lotus contends that the process to crack the hash is quite sophisticated and requires a very high access level to the system. The

specifics of this issue are not covered in the scope of this paper. *Computerworld* covered this issue in depth in this [article](#).

Can I See Some ID?

Before a client can be authenticated, its origin and validity must be verified. Just like a bouncer in a bar will question the teenager about his fake ID that says he's actually 44 years old and is from North Dakota, the Domino Server will make a client verify its credentials.

Notes uses the name, keys and certificates from the ID file to validate the identity of the user or server once the correct password for the ID file has been presented to the Notes client. As the Notes client application attempts access to the Domino server, it sends all of the certificates (both the user and ancestor certificates) from the ID file. The server then compares the ancestor certificates with the certificates in its own server ID file.

Here is the process:

1. John Doe/Accounting/Company X requests access to Server_A/Company X and sends all his certificates from his ID file to the server.
2. Because the server recognizes the common Domino Directory ancestor of Company X, it uses the Company X public key from its own ID file to validate that the organizational unit of Accounting/Company X was certified by its ancestor, Company X.
3. The process then reverses as the Server sends its certificates to the client, and the client validates the certificates of the server.

Your ID Looks Real, But Is This Really You?

Once the certificates of each entity have been validated, the client can be authenticated. The validation process only determines that the certificates for the organization can be trusted, but it does not prove identity. This is because a certificate associates the user with a public key for an organizational unit in the Domino Directory and tells the recipient that the public key can be trusted. However, in order to prove that user is really who he claims to be, he must show that he holds the private key that matches the public key in his certificate.

The actual process of authentication then occurs in a challenge/response format. Here is an example continued from above with John Doe accessing Server_A.

- John and Server_A exchange their validated public keys, and issue random number challenges to each other.
- Then, they exchange their complete list of certificates for each organizational unit and organization in their directory ancestries.
- Then, Server_A will send a secret key to John that will be used in a long term fashion for future interactions. Server_A encrypts this message with John's public key and applies its own private key to the message, effectively signing it and proving its origin.

- John then uses his private key to decrypt the long term secret key sent by Server_A. John then uses the long term secret key to encrypt its response to Server_A's original random number challenge and sends it back to Server_A, proving his identity.
- Once Server_A receives the response to its challenge that is encrypted with the long term secret key, it issues a session key back to John that will be used to encrypted data exchanges for this one session.

It is important to note that the authentication process is shortened if the two entities have previously validated their certificates and have already gone through the process of issuing and receiving the long term secret key. In subsequent interactions, the random number challenges are issued, and the client and server encrypt their respective responses using the long term secret key. Then the session key is issued and the session begins.

OK, So You Are Who You Say You Are, Now What?

Domino relies on authentication to differentiate one user from another, and employs access control mechanisms for every type of information in the Notes system. This means, once a session key is issued after authentication, the server checks that user's credentials in the Domino Directory for membership in access control lists, their specific access privileges, and their assigned user type for a resource. (Remember, in the Domino world, servers can be users of resources too.)

The Access Control List (ACL) for a resource is the definitive list of who has access to that resource, be it a single database or the configuration of a server. If a user or server is not on the ACL for an object, they are not allowed access to it. Access privileges govern user actions such as the ability to create or delete information or manage information through personal management scripts. User types allow the database manager to ensure that the type of ID file used for authentication must match the user type. A complete list of user types can be found in Appendix A.

The objects and process that can be restricted by the access control mechanisms are servers and ports, databases, design elements, and individual records which Notes refers to as documents. (Please note that this section assumes understanding of basic Notes design principles. The scope of this work is not intended to explain Notes database design.)

Servers and Ports

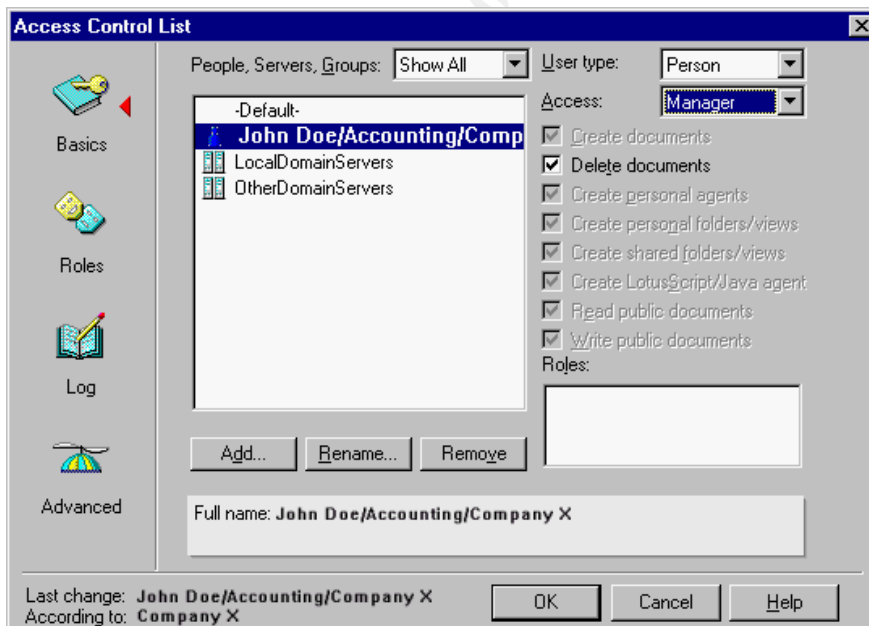
Access controls are employed to restrict access to specific network ports and user access to Domino server processes. Network ports for Notes can be actual protocols or TCP port numbers. For example, the domino administrator could restrict internal clients to a particular protocol like IPX. Server tasks such as creating new databases and configuring server operation can also be fully controlled by the administrator. Below is a screen shot from Domino

Administrator. Notice the tabs that allow for ports and server tasks security settings.



Databases

Every Notes database has an access control list (ACL) that specifies the level of access that users and servers have to the database. To control the access users and servers have to a database, the database manager specifies an access level and user type for each user in the ACL. Below is a screen shot of the ACL for a database:



In the center of the window for the ACL window is the list of the ACL members. Here John Doe/Accounting/Company X is shown as having the access level of manager. That means that he has all the access privileges shown with the

checkboxes on the right. Also note that he has the user type of Person. Access Levels are as follows: Manager, Designer, Editor, Author, Reader, Depositor and No Access. The full explanation of access levels and a list of the access privileges are given in Appendix B.

Another access control feature that is configured in the ACL window is User Roles. Below the list of access privileges is the list of roles for the selected user. In our example above, John Doe has no special role. However, by selecting the Roles icon on the left, the database manager can create roles and assign members of the ACL to that role. He or she can then use those roles to secure data design elements in the database.

Securing Design Elements

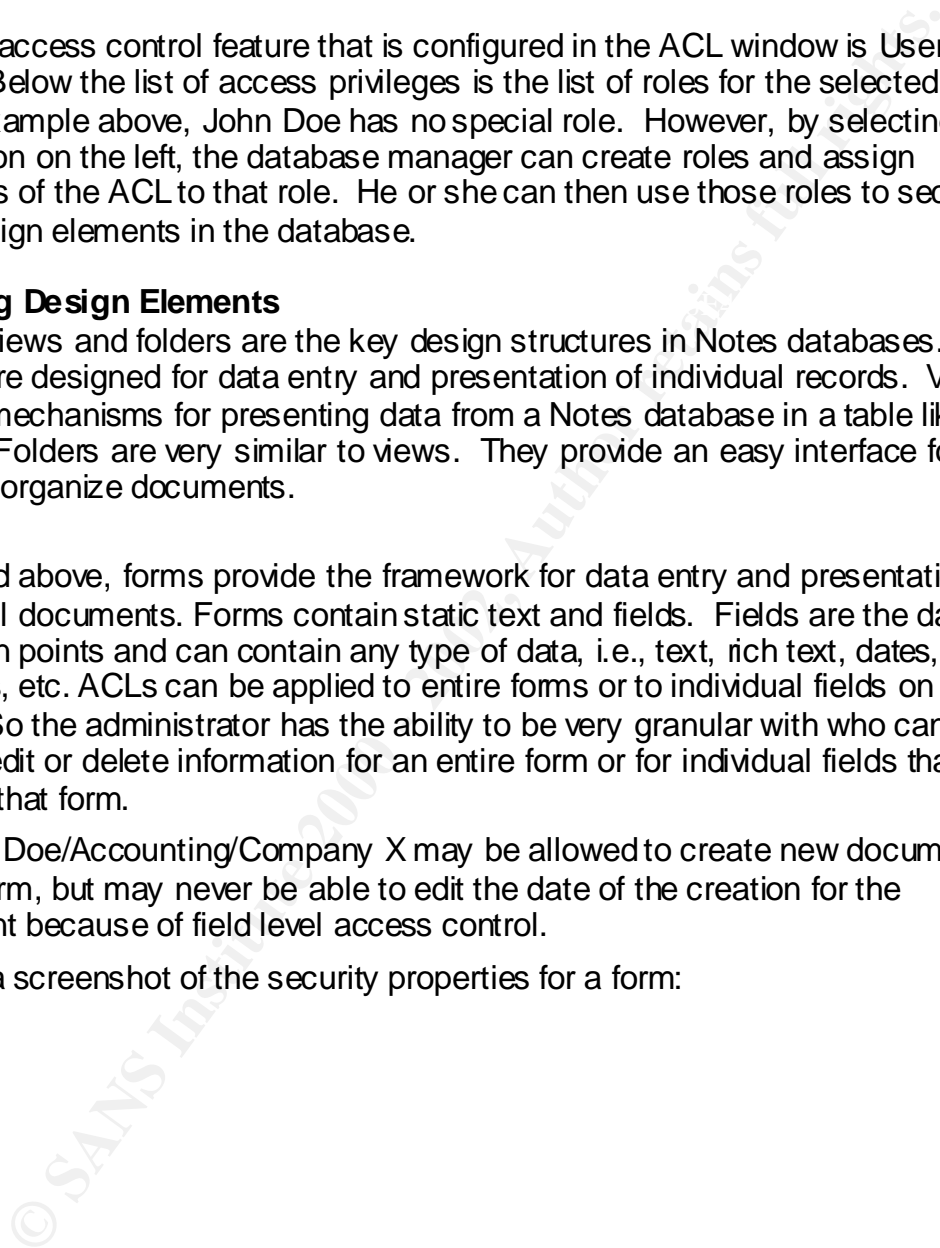
Forms, views and folders are the key design structures in Notes databases. Forms are designed for data entry and presentation of individual records. Views are the mechanisms for presenting data from a Notes database in a table like format. Folders are very similar to views. They provide an easy interface for users to organize documents.

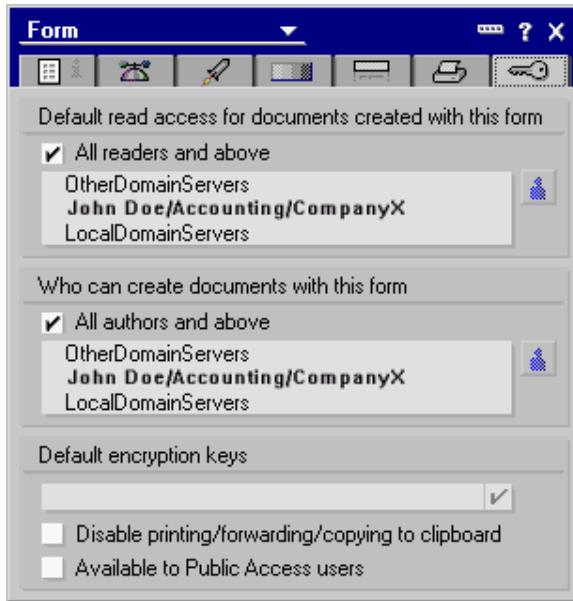
Forms

As stated above, forms provide the framework for data entry and presentation of individual documents. Forms contain static text and fields. Fields are the data collection points and can contain any type of data, i.e., text, rich text, dates, numbers, etc. ACLs can be applied to entire forms or to individual fields on forms. So the administrator has the ability to be very granular with who can create, edit or delete information for an entire form or for individual fields that exist on that form.

So John Doe/Accounting/Company X may be allowed to create new documents with a form, but may never be able to edit the date of the creation for the document because of field level access control.

Here is a screenshot of the security properties for a form:

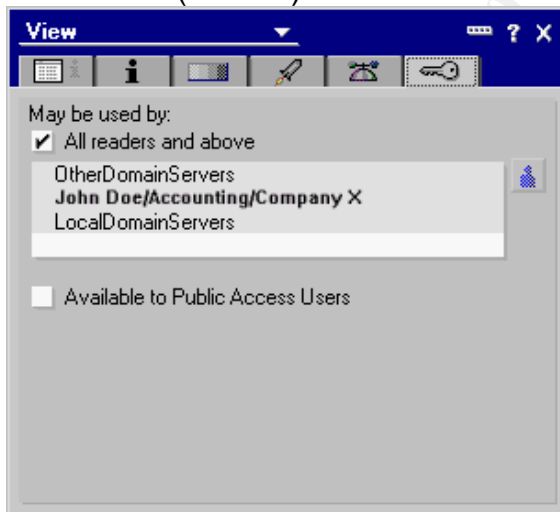




Note that the administrator can control who may read information with this form and who can create new information. Access is granted based on the access level specified by the ACL for the database, in this case, Readers and Authors.

Views and Folders

Views and folders are the mechanisms for presenting data from a Notes database in a table-like format. They allow users to see and access individual documents (records). Below are the security properties for a view.



Again note that the manager can restrict access to the view by the database ACL.

If the view in this example special information like a list of all transactions over \$500, then perhaps it would be restricted to John Doe in Accounting and not just anyone with Reader privileges.

Documents

As stated earlier, documents are the individual records in a Notes database. The Domino server administrator can restrict access to specific documents based on the ACL for the database, access privileges and user roles. An example of this type of security is to allow users who author documents in a database to see and edit only their own documents. That way, other users cannot alter their work.

Conclusions

Authentication and access control in the Notes/Domino environment is invisible to users but is very effective. The requirement of both a password and an ID file means that access to the system through a user account is much more than just cracking a password. The combination of certificates and encryption keys ensures the validity of user credentials and makes the forgery of ID files nearly impossible.

Access control in the Notes/Domino environment is pervasive and deep. Servers, design elements and database records can all be secured by employing Access Control Lists, access levels and privileges, and user roles. The flexibility of the access control mechanisms used by Notes/Domino allow system administrators to easily control access to the information in the system based on the user identity.

No computer information system is ever foolproof and Notes/Domino is no exception. However, its authentication and access control methods are well designed and comprise the foundation of very sound security model.

More security information on Notes/Domino is available on line at the following URLs:

<http://notes.net>

<http://www.lotus.com/security>

<http://www.searchdomino.com>

<http://www.redbooks.ibm.com/>

References

IBM International Technical Support Organization. *Lotus Notes and Domino R5.0 Security Infrastructure Revealed*. IBM Redbook SG24-5341-00, May 1999.

URL:

<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg245341.html>

(February 8, 2001).

Lotus Development Corporation. *Inside Notes: The Architecture of Notes and the Domino Server*. September 2000.

URL: <http://notes.net/notesua.nsf/find/inside-notes> (February 8, 2001).

Lowery, Tom. *Notes and Domino Defined: A Beginners Guide to Lotus Notes and Domino*. Domino Power Magazine, June 2000.

URL: <http://www.dominopower.com/issues/issue200006/notes001.html>
(February 8, 2001).

Disabatino, Jennifer. *Security experts warn of holes in Lotus Domino*. Computerworld, July 2000.

URL: http://www.computerworld.com/cwi/story/0%2C1199%2CNAV47_STO47808%2C00.html
(February 8, 2001).

Lotus Development Corporation. *Domino 5 Administration Help*. March 1999.

URL: http://doc.notes.net/domino_notes/5.0/help5_admin.nsf (February 8, 2001)

Lotus Development Corporation. *Security Glossary and FAQ*. Lotus Internet Security Zone.

URL:

<http://www.lotus.com/core/content.nsf/a1d792857da52f638525630f004e7ab8/a81d93e58a8612fd85256585005d0dc9?OpenDocument> (February 10, 2001)

Greenburg, Howard. *Designing A Secure Domino App*. Notes.net, June 1997.

URL:

<http://notes.net/today.nsf/cbb328e5c12843a9852563dc006721c7/71102330e24a7ce5852564b5005e3682?OpenDocument> (December 20, 2000)

Appendix A

User Types

(From *Inside Notes: The Architecture of Notes and the Domino Server*)

“A user type identifies whether a name in the ACL is for a person, server, or group. Assigning a user type to a name specifies the type of ID that is required to access the database with that name. The user types are Person, Server, Mixed Group, Person Group, Server Group, and Unspecified.”

Appendix B

Database Access Levels and Privileges

(From *Inside Notes: The Architecture of Notes and the Domino Server*)

Access level	Allows users and servers to
Manager	Modify the database ACL, encrypt the database, modify replication settings, delete the database, and perform all tasks allowed by lower access levels.
Designer	Modify all database design elements, create a full-text index, and perform all tasks allowed by lower access levels
Editor	Create documents and edit all documents, including those created by others. Read all documents unless there is a Readers field in the form. You must be able to read a document in order to edit it.
Author	Create documents only if the access privilege Create documents is selected. Edit the documents where there is an Authors field in the document and the user is specified in the Authors field. Read all documents unless there is a Readers field in the form.
Reader	Read documents. However, when the document contains a Readers field, only users whose names are listed in that field can read that document.
Depositor	Create documents.
No Access	None, with the exception of options to Read public documents and Write public documents.

© SANS Institute 2000 - 2002. All rights reserved.

Access privilege	Description
Create documents	<p>Determines whether a user can create documents in the database. If a user is listed in an Authors field of a document, the user can still modify that document.</p> <p>This privilege is automatic for Managers, Designers, Editors, and Depositors. It's optional for Authors.</p>
Delete documents	<p>Determines whether a user can delete documents in the database. If this privilege is deselected, a user can't delete documents, no matter what the access level.</p> <p>Authors can delete only documents they create. If the document contains an Authors field, an author can delete documents only if his user name is specified in the Authors field.</p> <p>This privilege is optional for Managers, Designers, Editors, and Authors.</p>
Create personal agents	<p>Determines whether a user can create personal agents in the database. Once created, a personal agent can perform only those tasks allowed by the user's assigned access level in the ACL. If the user creates an agent that runs on the server, the Agent Restrictions section of the Server document in the Domino Directory determines whether the agent can run.</p> <p>This privilege is automatic for Managers and Designers. It's optional for Editors, Readers, and Authors.</p>
Create personal folders/views	<p>Determines whether a user can create personal folders and views in a database on a server. Personal folders and views created in a database on a server are more secure than those created locally, and they are available on multiple servers. If the Create personal folders/views privilege is not selected, users can create personal folders and views and store them on their local workstations.</p> <p>This privilege is automatic for Managers and Designers. It's optional for Editors, Authors, and Readers.</p>
Create shared folders/views	<p>Determines whether a user can create shared folders and views in a database. Deselect this option to maintain tighter control over database design. Otherwise, users can create views visible to others.</p> <p>This privilege is automatic for Managers and Designers. It's optional for Editors.</p>
Create LotusScript/Java agents	<p>Determines whether a user can create LotusScript and Java agents in a database. Since LotusScript and Java agents on server databases can take up significant server processing time, database managers may want to restrict which users can create them. Whether or not a user can run agents is dependent on the access set by the Domino administrator in the Agent Restrictions section of the Server document in the Domino Directory.</p> <p>This privilege is automatic for Managers. It's optional for Designers, Editors, Authors, and Readers.</p>
Read public documents	<p>Determines whether a user can read public documents. This option lets you give users with no access the ability to view specific documents without giving them full reader access to the database.</p>

Note A document is public if it has a \$PublicAccess field with a text value of 1. Documents are not normally public; however, some specific documents -- such as, calendar and scheduling documents in a user's mail file -- are marked for public access.

This privilege is automatic for Managers, Designers, Editors, Authors, and Readers. It's optional for Depositors and No Access.

Write public documents

Determines whether a user can write public documents. This option allows users to create and modify documents with forms designed to allow public access. This option lets you give users create and edit access to specific documents without giving them Author access.

Note A document is public if it has a \$PublicAccess field with a text value of 1. Documents are not normally public; however, some specific documents -- such as, calendar and scheduling documents in a user's mail file -- are marked for public access.

This privilege is automatic for Managers, Designers, and Editors. It's optional for Readers, Depositors, Authors and No access.

© SANS Institute 2000 - 2002, Author retains full rights