Vivekanand R. Chudgar
CLP, PCLP, Solaris SA1, MCSE, MCP+I
GSEC Practical Requirements (v 1.3)

Security Features of Lotus Notes/Domino Groupware

Introduction

Lotus Notes has been one of the first complete groupware products to hit the market way back in 1989, and ever since it has continued to dominate the Groupware market. Developers of Notes realized the importance of Security quite early, and therefore we see many Industry Standard Security Features built into Notes over and above Security Features unique to Notes. Together, they effectively cover many aspects of Security that are of significant importance today.

Notes has a layered Security Model that provides Security Features at two broad levels:

1. Application Level – Features available to Notes Application Developers
2. Administration Level – Features available to Notes System Administrators

However, there's little information available on the scope of these security features (the aspects of security concerns they address) & therefore ***this paper aims to discuss the important Administration level security features of Notes with a focus on explaining the security they provide, their usage & also limitations/vulnerabilities.*** Application Level Security Features are not discussed here because it involves knowledge of coding & Notes Application Development skills, and also the scope of Administration Level Features itself is so vast that it merits a focused attention.

Various aspects of Security:

Notes addresses various aspects of Security. It is important that we understand these aspects of Security before moving on to discuss how Notes addresses them.

Following are the various aspects of security addressed by Notes [1]:

1. **User Security:** This refers to securing against the possibility of someone impersonating as another user and gaining unauthorized access to systems using someone else's credentials.

2. **Network Security:** This refers to securing against the possibility of the data

confidentiality & integrity being compromised while being transmitted over the Network.

3. **Database Security:** This refers to controlling access to each database such that only specifically authorized users can gain access to the data contained therein.

4. **Client Security:** This refers to securing the Notes Client against unauthorized access/intrusion.

5. **Server Security:** This refers to controlling access to each Notes server such that only the specified users can utilize the services provided by Notes Server (e.g. Access Databases, Create Replicas, Open Pass-thru connections etc).

Notes Features that address these Aspects of Security

Notes has many features, which address one ore more of the Security Aspects mentioned above (E.g. *Local Database Encryption* feature provides **Database Security** for Databases stored locally on a User's Desktop). Some of these important features are [2]:

Feature	**User Security**	**Network Security**	**Database Security**	**Server Security**	**Client Security**
Controlling Access using Access Control Lists			Yes	Yes	
Database Encryption	Yes		Yes		
Document Level Encryption	Yes		Yes		
Generate & Distribute your own Encryption Keys		Yes	Yes		
Signing Mails	Yes	Yes			
Encrypting Mails	Yes	Yes			
Encrypting Network Traffic		Yes			
Password Checking on Server	Yes			Yes	
SSL for Web Access	Yes	Yes			
Restricting local execution rights using ECL			Yes		Yes
Multiple passwords for ID files	Yes				
Increase delay between password prompts after every wrong password entry					Yes

Notes Components that provide these features:

Before we discuss in details these security features of Notes, it's essential to understand the basic components that Notes uses to implement these features. These are:

1. Domino Directory (NAB).
2. Notes IDs
3. Notes Certificates

The next section briefly explains these components with a particular focus on their security related roles/options.

1. Domino Directory: Domino Directory (Also referred to as NAB – Names and Address Book) is a special Notes Database that is critical to Notes operation. It holds all the critical configuration information about the Notes Environment implemented at the site (E.g. Users, Groups, Servers, Communication channels etc). It is created automatically at the time of configuring the first server in the Notes Domain & a replica of the same is maintained on all Servers created subsequently.

For each of the users, servers etc created in the Notes Domain, a document exists in the Domino Directory containing the configuration details about it [1]. Some of the configuration details pertain to access control, where by it specifies entities that are either allowed or disallowed to access this resource (E.g. *Deny Access List* configured in the Server Document for Each server). This helps in providing **Server Security** by controlling who can and cannot access server and perform certain actions on it [2] (E.g. create replicas, view Directory of Databases etc).

Vulnerabilities with Domino Directory: Domino Directory contains lots of system specific data and thus anyone managing obtain unauthorized access to it has the potential to do serious damage to all Notes Installations within the same Domain. Therefore, it's extremely important that the Access to Domino Directory is very tightly held, and monitored closely. Also, regular Backups of the Domino Directory are also very important to protect against any corruption/compromise of the same.

2. Notes ID: Notes ID is a unique file generated automatically by the system whenever a new User, Server or Certifier is registered. It contains:

1. The Name of the entity (User/Server/OU) that the ID file represents.
2. Information about the type of Notes License (N-American or International)
3. One or more certificates issued by one or more Certifiers.
4. Public & Private Encryption Keys
5. Other Encryption Keys

6. Password that is used to unlock the ID File & permits access to its contents.

The Notes ID files are referred to as User ID, Server ID or Certifier ID Files, based on the entity it represents. The ID file is a very important factor is Notes Security. If any user wishes to access any resource in the Notes domain, it must hold a valid ID File & present it to the system. The System then checks the ID File for valid certificates issued by the Certifiers from the same Organisation. Access is given to the user only when these certificates are checked and found OK.

Also, when a user wishes to use features such a Signing or Encrypting Data, the Private Key stored in the ID file is required to use these features. In addition to this, ID file also stores any Encryption Keys created by the user himself or provided to him by other users. Without these Encryption Keys, it will be impossible for the user to access the Data that is encrypted with these Encryption Keys.

Due to US Government restrictions, Lotus has come out with 2 different versions of Notes. They only difference between them is that they deploy different Key strengths [5].

North American	: 64 Bit Key
International/French.	: 40 Bit Key

Accordingly, the ID files & also the Notes Clients for these two versions are different. Also, French version also offers 40 Bit encryption, however, it is slightly modified by Lotus to meet the special requirements of French Government. For more details please refer to Article by Swedeen Bret: “Notes Encryption: Locks for a Digital World”

Vulnerabilities of ID Files: Since the ID file is so important; anyone managing to gain access to it can misuse it for accessing the Notes Domain. To prevent against the same, Notes provides for encrypting the contents of the ID file using the password provided by the user. Till such time that the correct password is provided to decrypt the contents of the ID file, it cannot be used to access the Notes Domain.

Also, another disadvantage of ID file is that if it’s lost, it cannot be recovered. In such cases, the user’s ID will have to be recreated. This is because of the fact that the ID file is the only place where a user’s Private Key is stored, & also the fact that it’s not possible to create the Private Key from the Public Key. To safeguard against such a possibility, most Organizations deploy Escrow Facilities where a safe copy of the Notes ID file is stored in a secure area. The same can be used in the event of the user loosing his ID file.

3. Notes Certifiers: Lotus Notes provides for Hierarchical Naming convention by use of Different Certifiers. All users and Servers must be certified by at least One Certifier in order to be able to access the various resources in the Notes Domain. When the first Notes Server is created, a single Top Level *ORGANISATION* is created which acts as the

Top Level Certifier. This Top Level Certifier can certify all Users and Servers, or alternatively, multiple *ORGANISATION UNITS (OU)* can be created under this Top Level *ORGANISATION* & be used for certifying Users and Servers.

For each of these *ORGANISATIONS (O)& ORGANISATION UNITS (OU)*, Notes creates an ID File with it's own certified Public & Private Keys. The access to the Certifier ID is also password protected and Notes ensures that the correct password is provided before the Certifier ID is used to certify any user.

Vulnerabilities with Certifiers: Since Notes certifiers control the creation of new users in the Domain, anyone managing to gain unauthorized access to the Certifier ID file has the potential to create additional users in the Domain and thereby compromise the security of the System. To protect against this possibility, Notes provides the facility of locking down the ID file with multiple passwords (Up to Four). This allows you to assign up to four different Administrators for the ID file, each having his own unique password to unlock the ID. All or some of the Administrators are required (based on the settings) to jointly enter their passwords before the ID file can be unlocked.

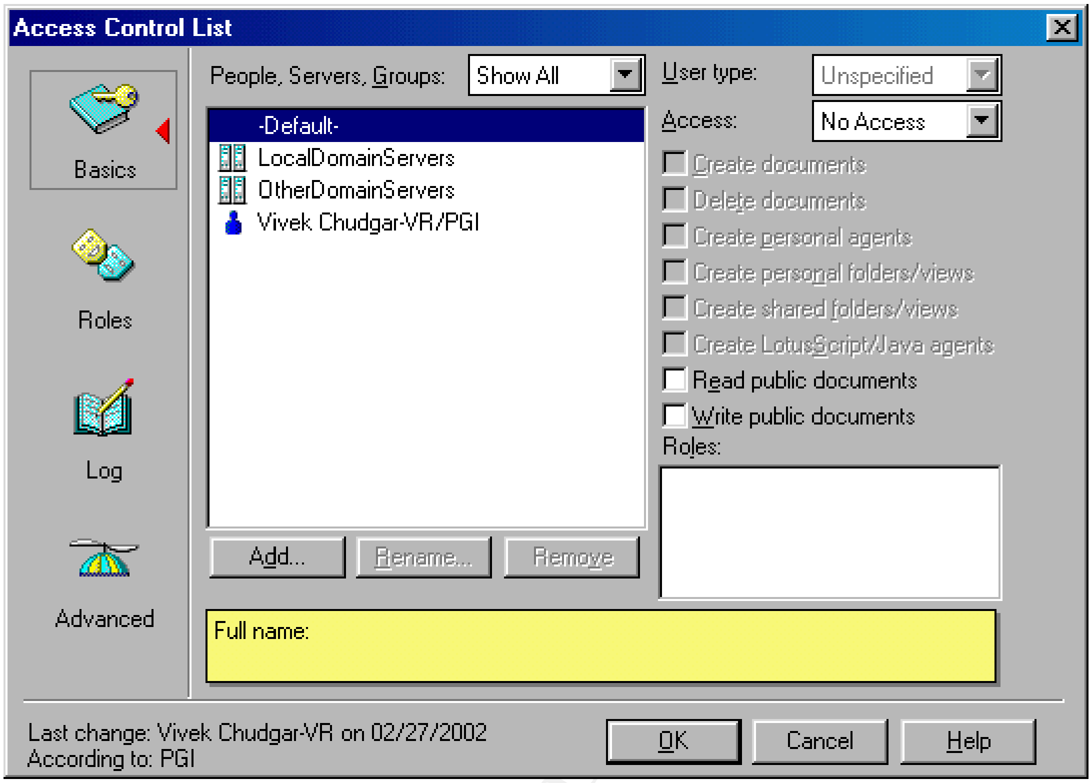
Now, with the overview of the Basic components of Notes over, it's time to explain how Notes provides the various Security Features using these basic components.

Notes Features that provide security:

1. **ACL - Access Control Lists:** ACLs provide one of the most significant ways in which the Access to the Data on any Notes Database can be controlled & restricted. ACL controls access to Notes Databases from everywhere (E.g. Notes Clients, Web Browsers, POP3 Clients etc). Each Notes Database has it's own ACL, and therefore, allows for setting individual access restrictions for each database over and above the access restrictions imposed by the Server Document.

Each ACL contains the following:

1. Names of Users, Servers and Groups that are allowed/denied access
2. Exact Type of Access allowed (E.g. Read Only/Author/Designer etc).
3. User Type (E.g. User or Server or Group)
4. Log of all the ACL changes made till date
5. Facility to Add/Remove/Rename various *Roles*
6. Name of Administration Server for the ACL/Database
7. Level of Access allowed to Internet Users



People, Servers, Groups: Show All

- -Default-
- LocalDomainServers
- OtherDomainServers
- Vivek Chudgar-VR/PGI

User type: Unspecified

Access: No Access

□ Create documents
□ Delete documents
□ Create personal agents
□ Create personal folders/views
□ Create shared folders/views
□ Create LotusScript/Java agents
□ Read public documents
□ Write public documents

Roles:

Full name:

Last change: Vivek Chudgar-VR on 02/27/2002
According to: PGI

Of these, the important points to note are:

Types of Access: ACL allows 7 different types of access levels to a Database [3]. The most restrictive is *No Access* & the least restrictive is *Manager*. *Manager* access is very important since it allows the user to edit data, make design changes & also modify the ACL for the Database. Therefore, you should ensure that its use is limited and given to only those who need to manage the Database and the ACLs.

Specifying User Type: Due to the architecture of Notes, and more specifically the feature of Replication, it becomes necessary to provide *Manager* Level Access to some Servers. This means if anyone is able to manage access to the Server ID, he can effectively gain *Manager* Rights to all the Databases for which this server is designated as *Manager*. Specifying the "User Type" helps to secure against such possibilities [3]. E.g. designating any ID as Server ID will prevent anyone from using that ID on a Workstation Client to access the Databases.

Level of Access to Internet Users: Serving any Data on the web increases it's vulnerability and hence Notes provides this facility of overriding & thus limiting access level to any user when he accesses the information from the web. E.g. if this Field is set to *Editor* Access, then even a User having Manager Access will get effective rights of *Editor* when he accesses the Database from a Web Browser.

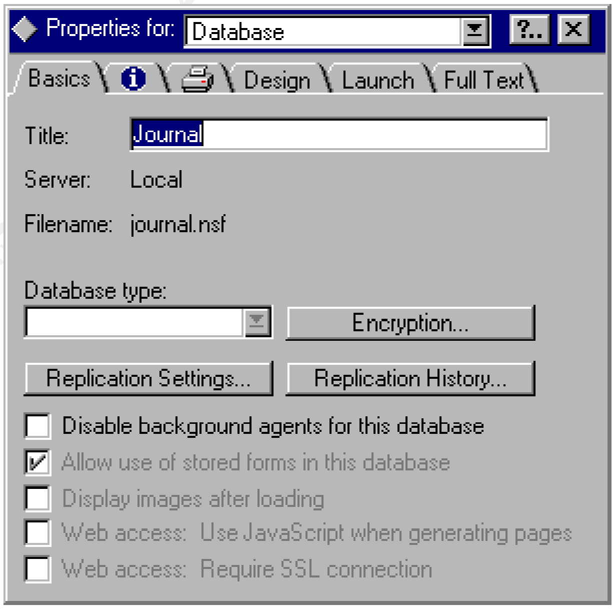
Limitation of ACL: A limitation of ACL is that a Notes Server is needed to enforce it. This means, for any Database that gets copied locally, the ACL is ineffective. However, other mechanisms are available for securing Databases in such a situation.

Also, one more limitation of ACL is that since it exists individually for each Replica of the same database, it is possible for each replica to have a different ACL. This could result in weak ACLs exposing that particular replica to the dangers of unauthorized access. To prevent this, & to also protect against lack of Access Control on Local Replicas, it's extremely important that the 'Maintain Consistent ACL across All Replicas' property is always checked [3] for all Databases unless specifically requested by the Developers due to their Design requirements. In such cases then the Administrator must take additional responsibility to ensure that tight ACLs are maintained across all different replicas.

2. **Database Encryption:** Database Encryption provides an additional level of security since ACL is not enforced on the Local Databases [5]. It can also be used for securing databases hosted on the Server, however it's mostly used for securing the local databases. Symmetric Key Encryption is used for this, where by a Random Encryption Key is generated and used to encrypt the Database. This Key is then encrypted with the Public Key of the user and attached with the Database.

To locally Encrypt a Database, select the Database, and click on *File > Database > Properties*. On the following Screen, click on Encryption & select the following:

- ID for which you want to encrypt the Database
- Strength of Encryption



Properties for: Database

Title: Journal

Server: Local

Filename: journal.nsf

Database type:

☐ Disable background agents for this database

☑ Allow use of stored forms in this database

☐ Display images after loading

☐ Web access: Use JavaScript when generating pages

☐ Web access: Require SSL connection

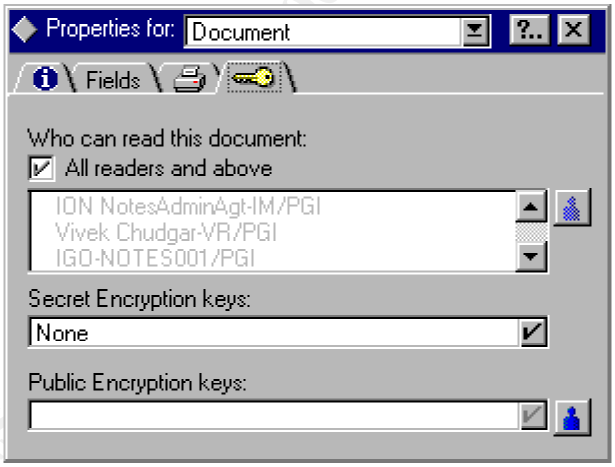
Selecting higher strength results in usage of a stronger Random Encryption Key for encrypting the Database. However, stronger key means more impact on the performance, since now more computation is required to recover the data from the encrypted Database.

Limitations: It is only possible to encrypt a Database with one user's Public Key. This means only one user can then access the Database contents. So if you encrypt a Database using someone else's Public Key, then you yourself will not be able to view that Database contents.

Also, you cannot use encryption keys generated by yourself for encrypting Databases. Therefore, the only option is to use Public Key of a user.

3. **Document Level Encryption**: This feature allows you to encrypt individual documents using Secret encryption keys (either generated by you or provided by someone else). Only users who have that particular Secret Encryption Key in their ID file can then access these documents. Therefore, if you generate the Secret Encryption Key, then no one else can access this document unless you share the Encryption Key with others [5].

To encrypt a Document, Click on *File* > *Document Properties* & go to the following screen. Note that the Option allows you to select the Encryption Key only from the list of available keys with you.



Properties for: Document

Who can read this document:

☑ All readers and above

ION NotesAdminAgt-IM/PGI
Vivek Chudgar-VR/PGI
IGO-NOTES001/PGI

Secret Encryption keys: None

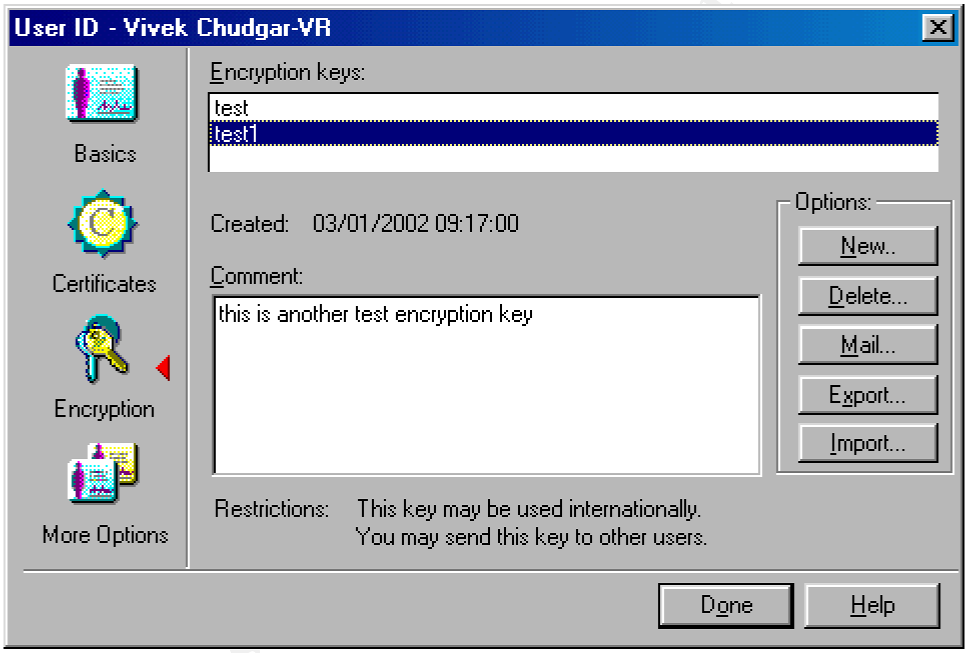
Public Encryption keys:

Please note that you can also encrypt a document using someone's Public Key available in your Address Book (or the Domino Directory). This would restrict the access to the document to only that user, since the document can now be decrypted only using that user's Private Key.

Limitations: Document Level Encryption is a powerful option. However, Notes does not provide any tool to encrypt multiple documents together. This means you are required to individually encrypt each document one by one. This severely restricts the ease of use, and hence very few users actually decide to use this feature.

4. **Generate & Distribute your own Encryption Keys**: Using this feature, you can generate your own Symmetric Encryption Keys and use them for encrypting Notes documents. These Keys are Symmetric Encryption Keys and hence provide quick Encryption but are not as secure as Asymmetric Encryption Keys. However, since Notes already uses Asymmetric Keys, the Symmetric or Secret Keys can be safely exchanged with others by encrypting it with the recipient's Public Key before transferring it over the Network [6].

To Create a Secret Encryption Key, Click on *File* > *Tools* > *User ID* & Provide your password when prompted. On the following screen, click on *Encryption* & Click on *New Key*. You will be prompted to enter a name and a brief description for the key, which will then be used to identify & refer to the key.


User ID - Vivek Chudgar-VR

Encryption keys:

test
test1

Created: 03/01/2002 09:17:00

Comment:

this is another test encryption key

Restrictions: This key may be used internationally.
You may send this key to other users.

You can now use this Key to encrypt any documents provided you have Author or Editor level access to that document.

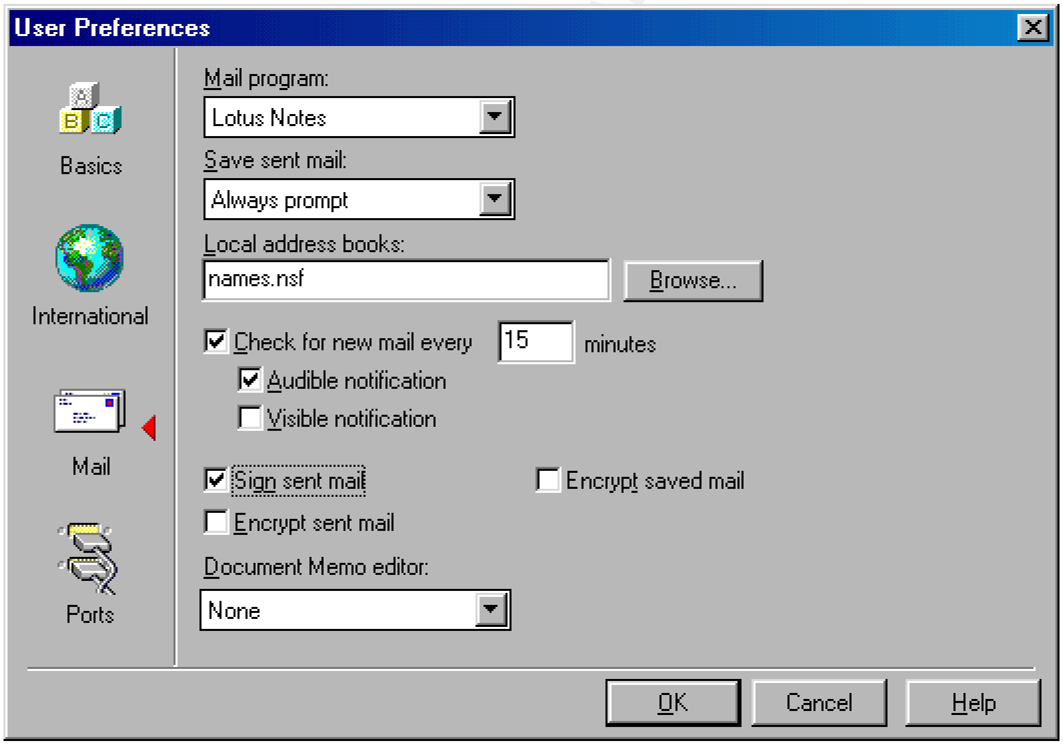
To distribute this Key, again go the screen shown above, Select the Key to be mailed & click on *Mail* Button. This prompts you for the Name of the recipient of the Key. The Key is then encrypted with the Public Key of the recipient, signed with the Private Key of the Sender, and mailed to the recipient. Encrypting with Recipient's Public Key ensures that only the Recipient can decrypt it (since Private Key is needed to decrypt this & only the Recipient's ID file has it). Signing with Sender's Private Key allows the Recipient to ascertain the identity of the Sender & ensure that the Key came from a valid source.

Notes also allows you to Export or Import the Key into a password protected & encrypted file. However, the onus of safely transporting it to the recipient then remains on you.

Limitations: Limitations of self-generated Encryption Keys are same as the limitations of Symmetric Encryption Keys (unsafe to transport & relatively easier to break). However, Notes takes care of the security while transport by using the already present PKI Infrastructure to securely transfer the Encryption Key between different users.

5. **Signing Mails**: This feature provides you the security of ascertaining that the sender of the message/data is actually who he claims to be.

When a user signs a message, Notes creates a Hash of the Message Body, encrypts the Hash with the Private Key of the Sender and attaches the Encrypted Hash along with the Mail. To sign a message, you can either click *Delivery Options* Button (on the Email) and checking the 'Sign' Option. Or alternatively, you could configure Notes to sign all mails sent from your Notes Client by clicking on *File > Tools > User Preferences* & checking the 'Sign Sent Mail' Option on the Mail Tab.



Mail program: Lotus Notes

Save sent mail: Always prompt

Local address books: names.nsf

☑ Check for new mail every 15 minutes

☑ Audible notification

□ Visible notification

☑ Sign sent mail

□ Encrypt saved mail

□ Encrypt sent mail

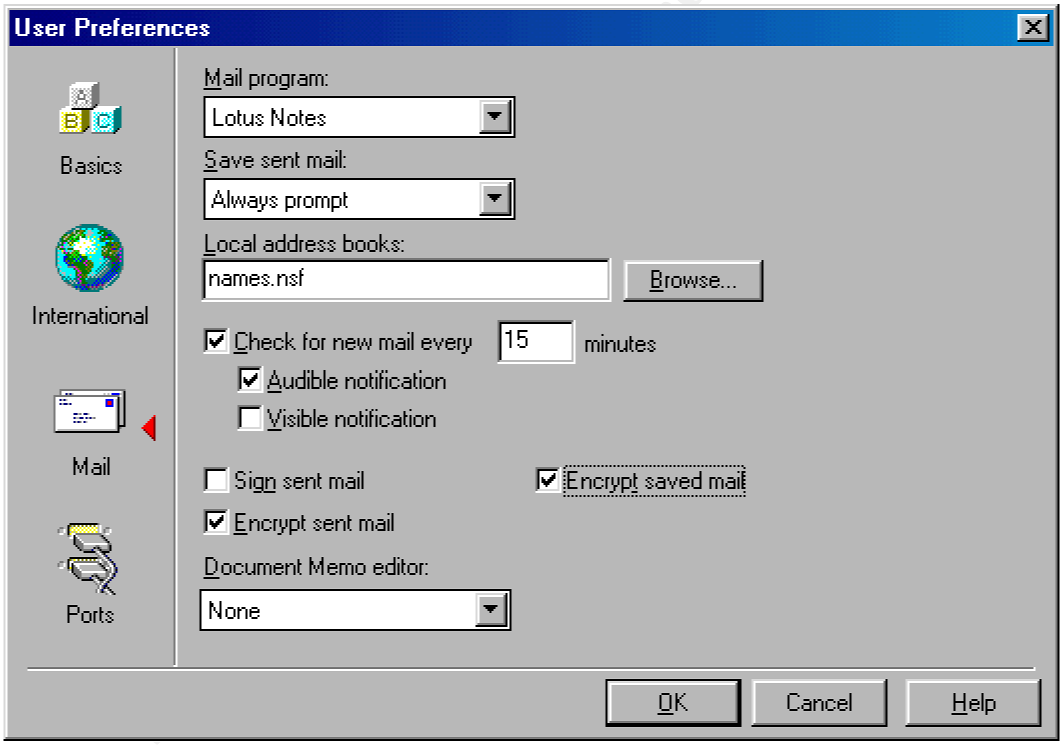
Document Memo editor: None

When the Recipient receives the Mail, he can ascertain the identity of the Sender by re-hashing the Message Body, & comparing it with the Hash received along with the Mail (After decrypting the Hash with the Public Key of the Sender). If there's any mismatch, then the Recipient will be prompted with the message that the Sign is corrupted.

6. **Encrypting Mails**: Encrypting messages provides the security of ensuring that the data remains safe while stored or transported and only the authorized users can access it.

When a user selects encrypting a Message, Notes generates a Random Encryption Key & encrypts the Message Body (not the To, Cc, Bcc & Subject Fields) with this Key [5]. Notes then encrypts this Random Key with the Public Key of the recipients and attaches it with the message.

To encrypt a message being sent, you can either click *Delivery Options* Button (on the Email) and checking the 'Encrypt' Option. Or alternatively, you could configure Notes to encrypt all mails sent or saved from your Notes Client by clicking on *File > Tools > User Preferences* & checking the 'Encrypt Sent Mail' and/or 'Encrypt Saved Mail' Option on the Mail Tab. It is also possible to encrypt all received messages by enabling 'Encrypt Incoming Mail' Option in each user's Person Document (in Domino Directory).



Mail program: Lotus Notes

Save sent mail: Always prompt

Local address books: names.nsf

☑ Check for new mail every 15 minutes

☑ Audible notification

□ Visible notification

□ Sign sent mail

☑ Encrypt saved mail

☑ Encrypt sent mail

Document Memo editor: None

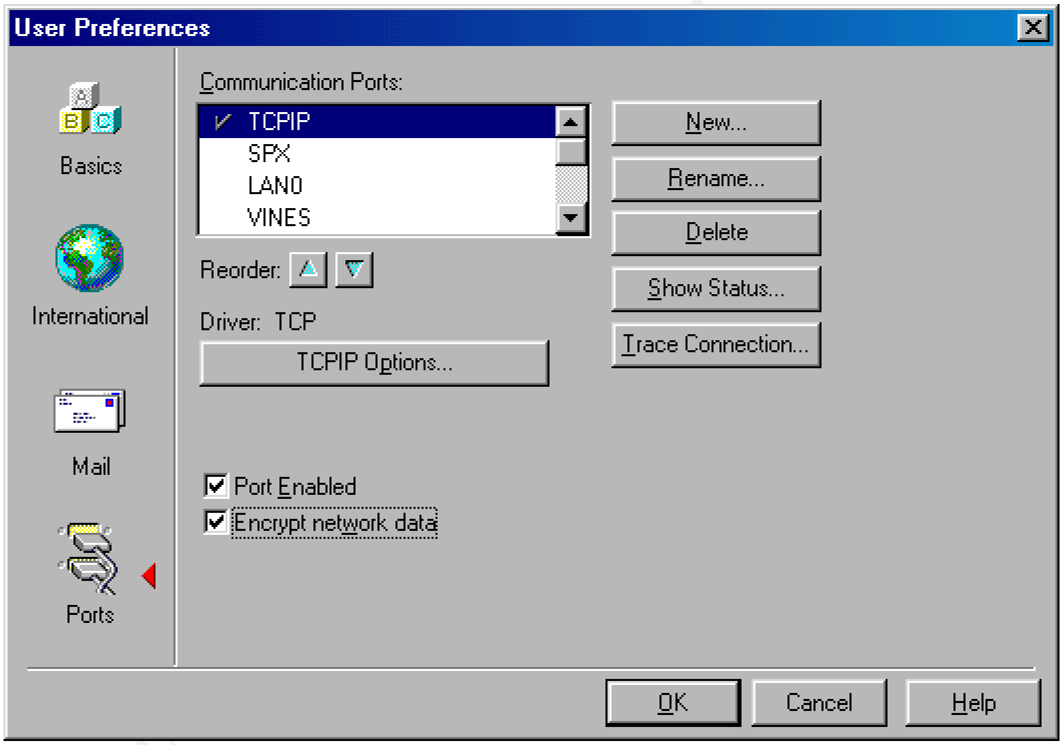
The recipient uses his Private Key to decrypt the Random Encryption Key, and once he has the Random Encryption Key, he can use it to decrypt the body of the message.

Notes also provides a special encryption option to encrypt all Incoming Mails of a user's mail file. This is also a very powerful security feature. It secures the mails against the threat of being read by Administrators who have Administrator rights over your mail file. To enable this feature, you need to open the user's Person Document (In Domino

Directory) and set the “Encrypt incoming Mails” property to *yes*. This will ensure that all Incoming mails get encrypted with your Public Key and hence only you can open them with your Private Key.

7. **Encrypting Network Traffic**: Encrypting messages only ensures that mail messages sent between the Notes client & the server is encrypted. However, it does not encrypt the traffic related to accessing other Notes databases hosted on other servers.

By enabling Notes to Encrypt Network Traffic, all the packets between two Notes boxes (Client–Server or Server–Server) are encrypted using a Random Encryption Key before sending over the wire [6]. To enable this feature for traffic between Notes Client and Server, click on *File > Tools > User Preferences*, go to ‘Ports’ Tab and check the ‘Encrypt Network Data’ Option.



Communication Ports:

✓ TCPIP
SPX
LAN0
VINES

Driver: TCP

☑ Port Enabled
☑ Encrypt network data

The Random Encryption Key is generated by the Server and sent across to the Client after encrypting it with the client’s Public Key. Once the Key reaches the Client safely, it’s used for all future data transfers till the session is closed. When the session is closed, the Key is discarded, and the next time a session begins, a new Key is generated again and the whole process begins from scratch.

Limitations: The resulting overhead of Network Data Encryption results in about 10% drop in traffic as compared to the Unencrypted Traffic. Also, if the client uses multiple ports to transfer data, then encryption must be individually enabled for each of these ports.

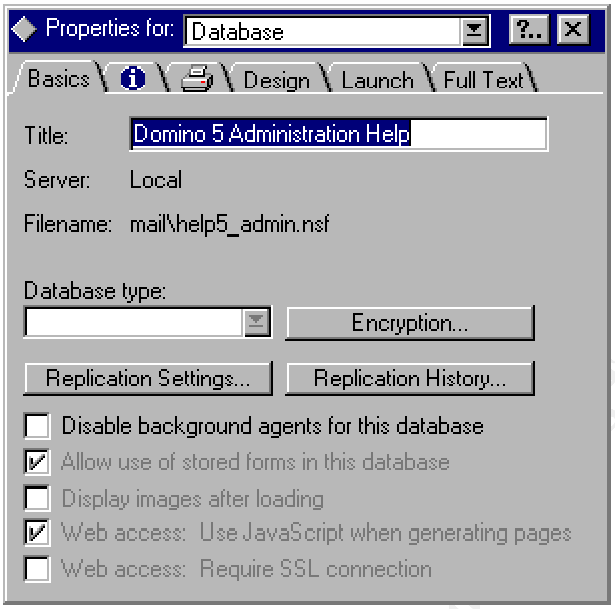
8. **Password Checking on Server**: Notes authenticates the users by checking the password locally. Therefore, if someone manages to get a copy of someone else's ID file and also learn the password of the same, he can access the Notes Domain as if he was the original user, without worrying about getting noticed/identified. The problem doesn't get solved even if the original user changes the password on his ID file, since the imposter has already got another copy of the ID file and the passwords are separately stored for each copy of the ID file.

The Option to check the password on the Server effectively eliminates this security breach [4]. With this option enabled, a copy of user's password is also stored in his person document in the Domino Directory. Whenever someone attempts to authenticate with a Server, Notes checks for his password by comparing it with the one stored in the Person Document of that user. If the password is different, then the access to server is denied to that user.

Please note that to enable password checking on server, you must enable password verification on both users and servers. This means, for each user that you wish to have password checking enabled, you must follow a process and enable that user for password checking.

Limitations: Password checking on server cannot be enabled for ID Files having multiple passwords.

9. **SSL for securing Web Access**: SSL Stands for Secure Socket Layer. This is an Industry Standard Authentication Mechanism that provides a reliable & secure channel of communication between a Web Server and a Web Client where the Web Client does not have it's own Public-Private Key pairs.



Title: Domino 5 Administration Help

Server: Local

Filename: mail\help5_admin.nsf

Database type:

☐ Disable background agents for this database

☑ Allow use of stored forms in this database

☐ Display images after loading

☑ Web access: Use JavaScript when generating pages

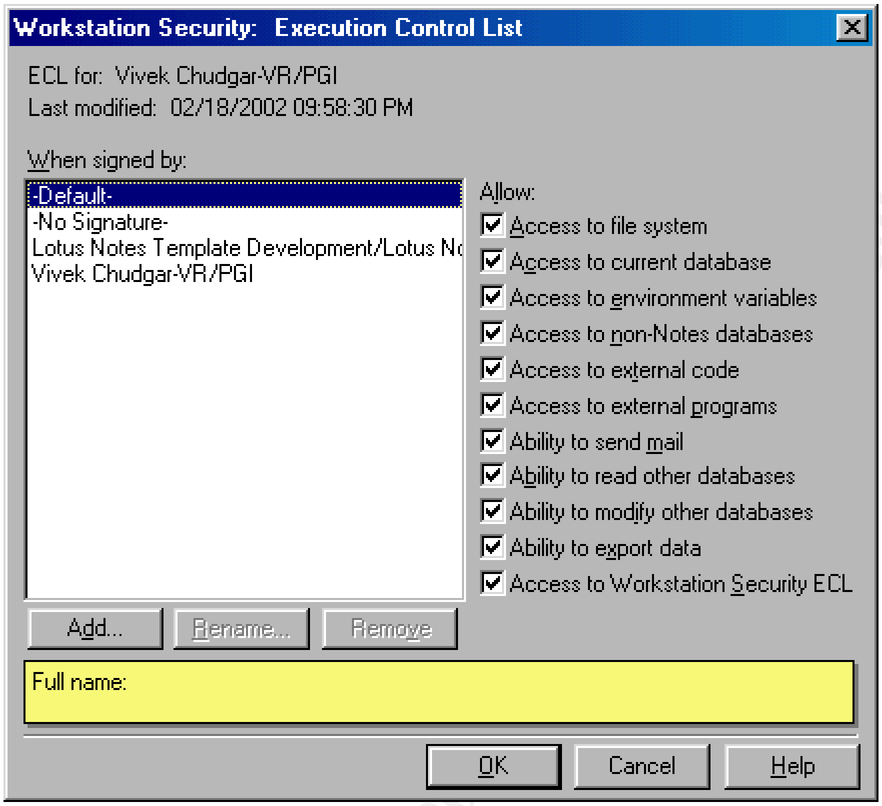
☐ Web access: Require SSL connection

You can setup SSL on Notes Server on protocol-by-protocol basis [8]. This means you can choose to enable SSL for POP3 & HTTP traffic only whereas the NNTP, SMTP & IMAP traffic flows normally. SSL gets configured individually for each Server. However, the databases continue to provide standard access to the data also. To force a database into only serving Data over a SSL connection, click on the following:

Notes allows the use of Certificates by commercial CA (such as Verisign) and also provides the facility of being your own CA and issuing your own Certificates to users/servers.

10. **ECL – Execution Control List**: ECL is a very powerful tool. It provides the facility to restrict the rights of locally executing any agents/actions created by others [9]. E.g. this allows you to restrict another person's code from executing on your Local Workstation, thereby avoiding the possibility of it damaging or erasing your Data. ECL also helps in reducing the risk of infection by malicious code [10] since it allows you to restrict who can execute a code locally on the Notes Client.

ECL is defined locally once for each Lotus Notes Client. It contains the Names of Users/Servers and the allowed activities that the user can perform on the Workstation.


Workstation Security: Execution Control List

ECL for: Vivek Chudgar-VR/PGI
Last modified: 02/18/2002 09:58:30 PM

When signed by:
- -Default-
- -No Signature-
- Lotus Notes Template Development/Lotus Nc
- Vivek Chudgar-VR/PGI

Allow:
- ☑ Access to 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

Full name:

Limitation of ECL: The only limitation of ECL is that it cannot be managed easily. When a Notes Client is set up on a Workstation for the first time, the ECL is copied from the Administration ECL defined in the Domino Directory. Subsequently, any changes that you wish to make to the ECL are not simple and easy to rollout to a large number of users.

It is possible to centrally change the Administration ECL and send an Email to users requesting them to click on a link (which activates a Notes Action of updating ECL), however this is cumbersome and also does not guarantee complete rollout since users who do not open/read/act on this Email will not receive the updated ECLs. This is probably the primary reason why very few Organizations actually deploy ECLs and exploit its benefits.

11. **Multiple Passwords for ID Files**: This feature is already explained while explaining ID files. Please refer to the Section of Notes ID files above.

12. **Increasing Delay between Password Prompts**: This is a unique feature of Notes that effectively safeguards against the brute-force techniques of breaking passwords. This feature is enabled by default and cannot be disabled.

With this feature, after every wrong password entry, the next password prompt is delayed longer. E.g. After 5 wrong password entries, the 6th password prompt comes up after over 60 seconds. This period keeps increasing exponentially with every subsequent wrong password. This makes is almost impossible to hack the password using even brute-force techniques, since it severely restricts the possibility of someone being able to try out thousands of passwords against the ID file even over a time span of a few days.

Conclusion: Notes is very evolved when it comes to addressing the security concerns of today's corporate world. We see many industry standard security technologies already built into Notes since a long time. Also, it's worth noting that Notes Team has made sincere efforts to integrate these technologies well with Notes & thus ensure that they can be used effectively without any major difficulties or loss of functionality.

There are many more features of Notes that contribute in making Notes a secure and reliable corporate Colaboration Tool. It was practically not possible to cover all of them in the scope of this paper. However, you can refer to the links bellow for wider and more in-depth knowledge regarding the other features of Notes security.

www.notes.net
www.notessecurity.org
www.lotus.com/security

References:

[1] "Inside Notes": URL: http://notes.net/notesua.nsf/find/inside-notes (10/11/2000)

[2] Bryant Susan & Williams Christie: Overview of Notes/Domino Security: URL: http://www.notes.net/today.nsf/8a6d147cf55a7fd385256658007aacf1/ed1d81a398e0bca385256abc00105f18?OpenDocument (09/04/2001)

[3] Slapikoff Rob & Lipton Russ: The ABC's of using ACLs: URL: http://www.notes.net/today.nsf/8a6d147cf55a7fd385256658007aacf1/be08e4acfc72cd72852565d9004cb61c?OpenDocument (04/01/98)

[4] Cornaia Mark "Password Checking": URL: http://www.notes.net/today.nsf/f01245ebfc115aaf8525661a006b86b9/55e4cbd0f3257be685256abc001a5c7c?OpenDocument (09/04/2001)

[5] Swedeen Bret: Notes Encryption: Locks for a Digital World: URL: http://www.notes.net/today.nsf/62f62847467a8f78052568a80055b380/443c6ed28496c1e7852566090060ca17?OpenDocument (06/01/98)

[6] Notes/Domino Security Infrastructure Revealed: http://www.redbooks.ibm.com/redbooks/SG245341.html

[7] Domino 5.0.2 Administration Help: URL: http://www.notes.net/notesua.nsf/0b345eb9d127270b8525665d006bc355/ee618fcf799109ef8525683200531b21?OpenDocument (11/23/1999)

[8] Calabria Jane & Kirkland Rob: “Professional Developer’s Guide to Domino”: QUE Corporation: 1997

[9] Smith, Amy: “Staying Alert with ECLs”: URL: http://www.notes.net/today.nsf/8a6d147cf55a7fd385256658007aacf1/3a9da544637a69b2852568310078b649?OpenDocument (12/01/1999)

[10] “Lotus Notes & Domino reduce the risk of Virus Attack”: URL: http://www.lotus.com/developers/itcentral.nsf/wdocid/67BEC14A50E33DE9852568E400604A0A?OpenDocument