

Lotus Notes And Domino Contribution to the HCI and Security Systems Workshop

Dave Wilson

IBM Software Group
5 Technology Park Drive
Westford, MA 01886 USA
+1 978 399 6371
davew2@us.ibm.com

Mary Ellen Zurko

IBM Software Group
5 Technology Park Drive
Westford, MA 01886 USA
+1 978 399 6389
mzurko@us.ibm.com

ABSTRACT

New end users of the Lotus Notes client and administrators of the Domino server typically have no background in security software. Interfaces to the end user and administrator security features have been modified to make these features accessible and scalable.

INTRODUCTION

Lotus Notes and Domino is a client/server groupware and application development product with security features including authentication, access control, encryption, and certification. Collaboration is the primary the end user activity supported by Notes. Rarely does any previous software experience prepare an end user or administrator for understanding, assimilating, and managing the security features of the product.

A security infrastructure has been core to Lotus Notes and Domino since its introduction. Work continues to be performed on this infrastructure as security capabilities and features are continually being offered to the client, server, and application developer. Consequently, the product developers have deep experience on embedding security in collaborative applications and infrastructure and using that application pull to deploy sophisticated security technology. For example, Public Key Infrastructure (PKI) was part of Notes from the beginning. Notes Remote Procedure Call (RPC) authentication is public key based, and public/private key pairs are disseminated to and used by every Notes client user. This deployment was accomplished at a time when PKI was still a novel research topic and continues into today when some pundits claim that PKI is intractable and unusable. Notes is widely believed to have the largest deployed PKI in the world.

Lotus Notes and Domino developers with backgrounds and interest in security are responsible for the security interfaces both for both the Notes end user and the Domino security administrator. Efforts to improve usability of the security features began in Version 4 and have continued through Version 6 (the current version). Over this time additional sophisticated security features, with their own usability requirements, have been added to Notes and Domino. Further, because of the popularity of the product,

Domino security administrators have been required manage an exploding user and server population. We propose to share both the design principles and lessons learned in developing effective and scalable security interfaces for Notes and Domino.

THE END USER EXPERIENCE

Over the lifetime of Notes and Domino the operating environment changed (from proprietary networks to the internet), certificate management matured (hierarchical certificates have replaced flat certificates), and threats became more sophisticated (viruses and other active content). Corresponding security features added to the Notes client by different developers at different times resulted in divergent interfaces scattered throughout the product. An end user went to one place to request an update to his certificates, another place to set his encryption preferences, and a third to specify trusted content signers. The security of the Notes client appeared overwhelming because the features were not presented in a single consistent interface.

The User Security panel -- a single access point to all Notes client security features -- was introduced in Version 6. Its design principles include detailed explanations of the security principles and features and care in placing esoteric and deprecated functionality in secondary panels (accessible from well labeled buttons). Further, the User Security panel is extensible so that the security settings of other installed Lotus products could be accessed from its interface.

Notes client support for active content (scripts and executables) in Version 4 required that end users make decisions about whose software they would trust to run on their machines. For example, when the user ran an application he might be prompted to allow some script to have access to a range of functions -- from access to the current document to write access to the file system. The information displayed in the initial design of the prompt was only the requested access and the script's signer's name (provided that the content was signed at all); typically, the user dismissed the prompt allowing the access to the current script and all subsequent scripts authored by the

signer. Recognizing the danger of this conditioned user response, the prompt was redesigned to include more background information about the intent of the script and the impact its execution. Further, based on the usability evaluation of the process of users making active content decisions [1], new functionality was provided to allow end user to defer these decisions to the administrator.

Experience with the user community is reflected in the metamorphosis of the Notes client security features: logical presentation of all available security features and detailed explanation of the security concepts represented by the features from within the interface enables the end user to make better security choices; allowing administrators to easily set security parameters for end users eliminates errors.

While some security features either have a dedicated area in the product's user interface or are made evident to users via modal dialogs, others are so elegantly integrated into the user experience that the user doesn't even know they exist (and may not take advantage of them as a result). Minimizing user experiences where the security features like the PKI infrastructure could intrude has been a win in terms of deployment, but recent information indicates the security features may be cloaked too well. For example, while the signature status of a document appears in our status area, and can be double checked via a menu query, anecdotal evidence indicates that many users are unaware that Notes signatures are used by their corporation on important electronic documents, such as pay stubs.

THE ADMINISTRATOR EXPERIENCE

Access to the administration features (adding users for example) is typically controlled by some secret known to a limited number of trustworthy people. For the Domino security administrator, this secret is the location and the password of the organization's certifier id file. As the number of users of the system grows, more administrators typically need access to the security features (and the associated secret) in order to keep up with the workload. If one of these administrators proves untrustworthy (perhaps leaving the company in possession of the secret), the security of the system is compromised.

Version 6 of Domino eliminates this danger with the introduction of the Certificate Authority server process. The process configuration allows many administrators to be given the rights to access security features associated the certifier id file without giving them physical access to it. Another feature that can control damage by an untrustworthy administration was introduced in Version 5 of Domino: multiple passwords on an id file. A certifier id file can be configured to need more than one password to unlock it; consequently, an untrustworthy administrator acting alone cannot cause any damage even if he manages to get a copy of the organization's certifier id.

Additional flexibility is required when tasks originally designed to be an administrator or management function

are determined to be appropriate to for end users. For example, the initial structure of our Access Control List permissions was totally task oriented. A small number of access levels (such as Reader, Author, Editor, Designer, Manager) are tuned to expected use. This design simplifies the process of bootstrapping access control to a database and the approach has been quite successful overall. However, since our early versions, customers have asked to have some ability of make limited fine-grained tweaks to those levels. As an example, some enterprises want to give users full control over the access control of their mail database (a manager level permission), but don't want to let them be able to delete it. Their experience had show that users were prone to mistakenly deleting their mail database, substantially increasing the cost of their help desk and administrative operations (not to mention user frustration).

In early versions of Domino security attributes associated with individual users (how often they must change their password, for example) had to be explicitly set on the targeted users. When a new user was added to the system, an additional step was required to set his security attributes. When a large number of users was added to an organization Domino security administrators often forgot to set their security attributes. Version 6 of Domino introduced the Policies feature where attributes (one set of them being security related) can be assigned to an end user implicitly based on the organization to which he belongs.

As a system scales to include more manageable objects that need security configuration, introduction of delegation and cooperation concepts into the security administration interface provides protection against internal maliciousness and flexibility. Having security attributes implicitly assigned to an object when it is entered into a system provides a baseline for keeping the system secure. The system should allow for exceptions to implicitly assigned attributes for flexibility.

THE DEVELOPER EXPERIENCE

Security is the primary design goal of developers of security software; other attributes of the software (usability, for example) have relegated importance. End user security features are often infrequently accessed so their usability evaluation lags behind that of more general functionality. Further, because security concepts are not immediately accessible, a usability engineer or QA engineer will often defer to the security developer regarding designs and interfaces. Sadly, security features are released with user interface issues and the issues are addressed only when customers complain.

One example of a security feature slipping through the usability cracks is password quality. Notes client users must provide a password to open their id file. Administrators can specify that the password must have at least a certain number of characters, or more securely, that the password be of a particular quality (on a range of 0 to 10) -- the higher the password quality the more difficult it

is to guess. The first interfaces for setting password quality (for administrators) and for specifying a password of a particular quality (for the end user) displayed definitions of different levels of password quality (easy to guess, hard to guess, etc.) but gave no explanation of the characteristics of a password of a particular quality and no examples of passwords that met the quality criteria. Security conscious administrators directed their users to create password of high quality, and users struggled to find a password that matched the rules (increasingly frustrated users as reported in [2]). Thankfully, subsequent versions of the password quality user interface gave hints on how to create passwords of a particular quality and examples.

One design principle of security software is to get acknowledgement from the end user before performing a function on their behalf (as a protection against being spoofed). Presenting an interface to the user can disrupt the effectiveness of the administration of a security system however, if the security administrator requests the operation and the user does not execute it. Version 5 of Domino included a feature that escrowed modified user id files to a secure repository; the id files could then be accessed if the user lost their copy of the id file or if they forgot their password. Although the repository's email address was specified by the administrator when he enabled the function, the user whose id file was being escrowed was presented with a mail address dialog when they modified their id file (and subsequently invoked the function). The user could modify the address (having the id file routed somewhere other than the repository) or not send it at all. Asking for user acknowledgement and input in this instance defeated the purpose of the feature.

Security features may need configuration to work correctly. Well designed security systems will check for mis-configuration and will report issues to the security administrator. Domino provides a document level security feature where the ability to edit or see the document relies on a person's name being included in one of the document's fields. When a person's name is changed, their name will be changed in these document fields if the Notes database is properly configured; if not, the name is not changed and the user loses access to the document. While it is not true that all databases should be configured to maintain these fields, the software should detect when a candidate database is not configured and report the condition to the administrator.

A final desirable attribute of a security system is its programmability. Regardless of any efforts to improve the usability of security interfaces (especially for administrators) some organizations prefer to integrate the security features of software applications into their own customized interfaces (either as part of a workflow or to pull all security operations into one interface). Domino's programmability via various scripting languages and APIs

allows access to its administration security features from other interfaces.

Developers should recognize making assumptions about user response, proper configuration, and functional accessibility may decrease the usability and effectiveness of the security features they deliver.

OPPORTUNITIES

No system is perfect, so opportunities still exist to improve the usability of the security features available in Notes and Domino. A few of these ideas are briefly presented below.

Security help desk -- when an end user needs assistance from the Domino security administrator (for example, because he's been prompted to do something but does not know where the interface to the function is), the question is typically asked and answered via the telephone or email. An improvement would be an interface where the user could post his question or issue, receive information from the security administrator, and see questions that other users have asked. Providing this type of functionality would help in educating end users about the security features of the product.

Administrator security panel -- Domino security administrators are often trained network administrators who have been asked to take on additional responsibilities. While some of the concepts of secure network administration have parallels in Domino security administration, others do not. An administrator security interface that used the same design principles as the User Security Panel would be effective in bringing these converted administrators up to speed.

Security Administration workflow -- when Domino security administrator performs a user management operation (like name change or recertification) logging the events that the administrator requested the change and the Notes client registered it would be useful in tracking down configuration problems.

References

1. Mary Ellen Zurko, Charlie Kaufman, Katherine Spanbauer, Chuck Bassett. Did You Ever Have To Make Up Your Mind? What Notes Users Do When Faced With A Security Decision. <http://www.acsac.org/2002/papers/7.pdf>
2. Yan, Jianxin Jeff. A Note on Proactive Password Checking, in *Proceedings of New Security Paradigms Workshop 2001* (Cloudcroft NM, September 2001), ACM Press, 127-135.