



AdminCamp 2014 Track 2, Session 3:

Weil sicher sicher sicher ist - Sicherheit in IBM Domino

Gelsenkirchen, 30. September 2014

Innovative Software-Lösungen.

www.assono.de



Thomas Bahn

Diplom-Mathematiker, Universität Hannover

seit 1997 entwickle ich mit Java und relationalen Datenbanken

seit 1999 mit Notes/Domino zu tun:
Entwicklung, Administration, Beratung und Schulungen

regelmäßiger Sprecher auf nationalen und internationalen Fachkonferenzen zu IBM Lotus Notes/Domino und Autor für THE VIEW



 tbahn@assono.de

 <http://www.assono.de/blog>

 04307/900-401





Agenda

- Worum geht es in dieser Session **nicht**?
- Berechtigungen – vom Netzwerk bis zum Abschnitt
- Pragmatische Einführung in die Kryptographie
- Zertifikate und ID-Dateien
- Verschlüsselung – vom Netzwerk bis zum Feld
- Signaturen
- Authentifizierung
- Passwort-Management
- Sicherheit in Notes-Anwendungen (wenn noch Zeit bleibt)



Worum geht es in dieser
Session nicht?



Worum geht es in dieser Session **nicht**?

- Safety vs. Security
- Datensicherung, Verfügbarkeit
- Sicherheit von Web-Anwendungen und Internet-E-Mails
 - SSL, S/MIME, ...
- Sicherheit von Notes-Client-Plug-Ins, Widgets, Composite Apps
- Physische Sicherheit
- Betriebssystem, Dateisystem, Netzwerk, Anti-Virus...
- weiter-/tiefergehende Details



Berechtigungen – vom Netzwerk bis zum Abschnitt



Berechtigungsebenen

- Netzwerk
- Server
- Verzeichnis
- Datenbank
- Dokument
- Kontrollierter Zugriff-Abschnitt



Netzwerk

- Allow_Access_<port> in notes.ini
www-10.lotus.com/ldd/dominowiki.nsf/dx/allow_access_portname
- Deny_Access_<port> in notes.ini
www-10.lotus.com/ldd/dominowiki.nsf/dx/deny_access_port
- Beispiel:
Allow_Access_ClusterTCPIP=ClusterServerUndAdmins
Deny_Access_TCPIP = AusgeschiedeneMitarbeiter



Server-Dokument – Sicherheit – Administratoren

- Administratoren mit voller Berechtigung:
 - nur Personen
 - Ereignis-Generator und -Handler in events4.nsf für Benachrichtigung, jedesmal wenn aktiviert
 - hilft nicht gegen Verschlüsselung
- Administratoren: LocalDomainAdmins
- weitere Felder werden eher selten verwendet
- Administratoren mit voller Remotekonsolen-Berechtigung:
LocalDomainServers oder Agent-Signer-IDs, wenn Agenten Serverkonsolen-Kommandos nutzen können sollen
(NotesSession.SendConsoleCommand)



Server-Dokument – Sicherheit – Programmierbarkeit

- Wer darf Code ausführen
 - Formelsprache
 - Beschränktes LotusScript/Java
 - Unbeschränktes LotusScript/Java (OS-Befehle!)
- Agenten signieren, die im Namen anderer ausgeführt werden, Agenten oder XPages signieren, die im Namen des Aufrufers ausgeführt werden, Scriptbibliotheken signieren, die im Namen anderer ausgeführt werden und ggf. Unbeschränkte Methoden und Operationen signieren oder ausführen:
nur Personen



Server-Dokument – Sicherheit – Sicherheitseinstellungen

- Public Key Checking
 - Öffentliche Schlüssel vergleichen: Überprüfen von öffentlichen Schlüsseln für in vertrauenswürdigen Verzeichnissen aufgeführte Notes-Benutzer und Domino-Server zwingend
 - Nichtübereinstimmungen von öffentlichen Schlüsseln protokollieren: Nichtübereinstimmungen von öffentlichen Schlüsseln für alle Notes-Benutzer und Domino-Server protokollieren
 - mindestens protokollieren, um zu korrigieren
 - sonst kann Benutzer für ihn verschlüsselte Informationen (wie E-Mails) nicht lesen
- Anonyme Notes Verbindungen zulassen: Nein
- Kennwörter von Notes-IDs überprüfen: Aktiviert



Server-Dokument – Sicherheit – Internetzugriff

- Internet-Authentifizierung:
Weniger Namensvariationen mit höherer Sicherheit



Server-Dokument – Sicherheit – Auf Server zugreifen

- Serverzugriff:
In allen vertrauenswürdigen Verzeichnissen aufgeführte Benutzer und Server-Name und LocalDomainServers
- Kein Serverzugriff:
Ausgeschiedene Mitarbeiter und Praktikanten



Server-Dokument – Sicherheit – Auf Server zugreifen

- Datenbanken und Schablonen erstellen:
LocalDomainAdmins, LocalDomainServers
- Neue Repliken erstellen:
LocalDomainAdmins, LocalDomainServers
- Masterschablonen erstellen:
LocalDomainAdmins, LocalDomainServers
- Verwendung von Monitoren zulässig für:
LocalDomainAdmins
- Vertrauenswürdige Server:
LocalDomainServers



Server-Dokument – Ports... – Internet-Ports

- SSL-Verschlüsselungscodes:
AES aktivieren, alles mit MD5 MAC und DES deaktivieren,
ggf. Triple-DES und RC4 deaktivieren
- Einstellungen zum Serverzugriff erzwingen:
Ja (für alle Protokolle)
- Alle Ports deaktivieren, die nicht benötigt werden – verhindert
Prozessstart „aus Versehen“



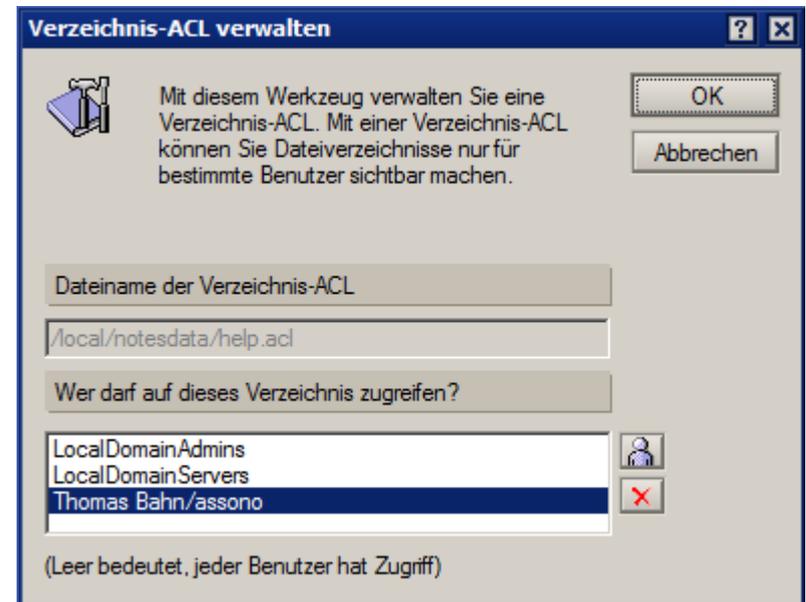
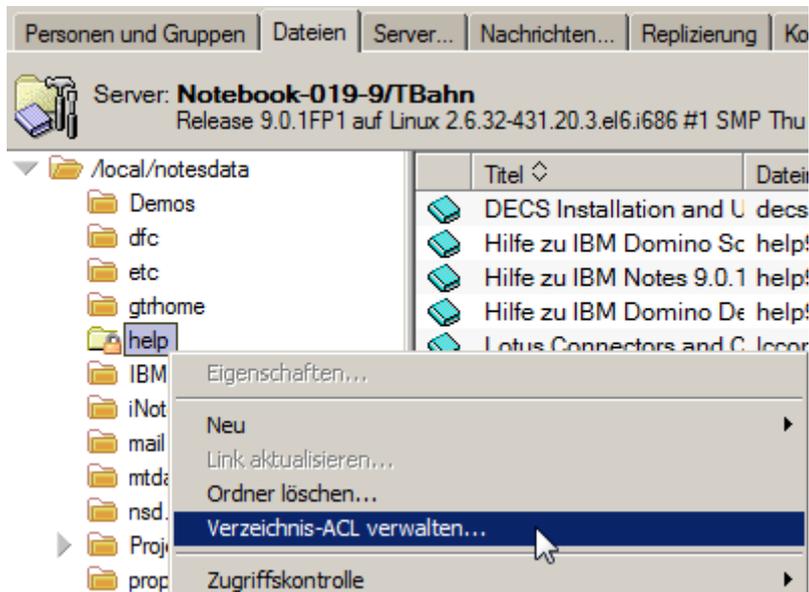
Server-Dokument – Notes Traveler

- Zugriff auf IBM Notes Traveler
- Serverzugriff:
 - In allen vertrauenswürdigen Verzeichnissen aufgeführte Benutzer
- Kein Serverzugriff:
 - Ausgeschiedene Mitarbeiter und Praktikanten



Verzeichnis

- Voraussetzung: Enable_ACL_Files=1 in notes.ini
www-10.lotus.com/ldd/dominowiki.nsf/dx/enable_acl_files





Datenbank

- Zugriffskontrolle = Access Control List = ACL
- 7 Berechtigungsstufen
 - Manager
 - Entwickler
 - Editor
 - nur in Ausnahmefällen sinnvoll
 - Felder konfigurierbar: nur für \geq Editoren änderbar
 - Autor
 - für mich einzige sinnvolle Option für alle Server und Anwender außer Administratoren
 - Leser
 - Einlieferer
 - Kein Zugriff
 - Sollwert für -Default- und ggf. Anonymous



Datenbank (forts.)

- Weitere Berechtigungen
 - Dokumente erstellen
 - Dokumente löschen
 - Private Agenten erstellen
 - Private Ordner/Ansichten erstellen
geht aber immer!
 - Gemeinsame Ordner/Ansichten erstellen
 - LotusScript/Java-Agenten erstellen
 - Öffentliche Dokumente lesen
 - Öffentliche Dokumente schreiben
 - Dokumente replizieren oder kopieren
gilt auch innerhalb von Dokumenten
und für das Drucken!
- Rollen



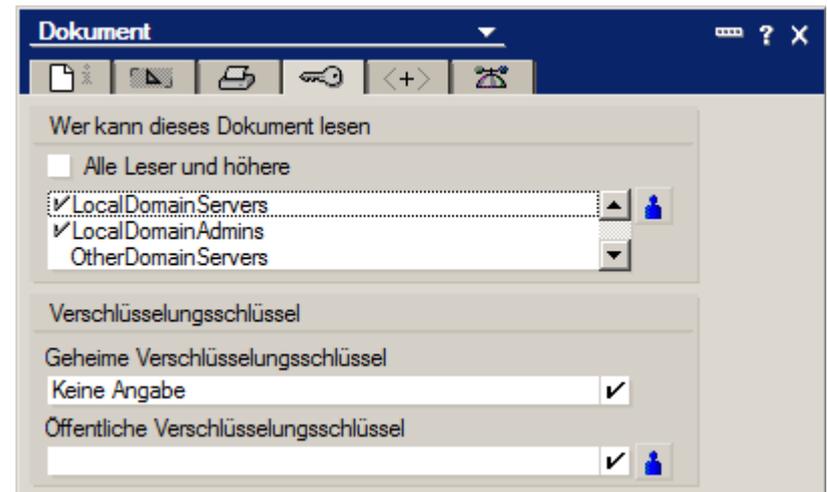
Datenbank

- ACL – Erweiterte Einstellungen
 - Administrationsserver
immer setzen!
 - Konsistente ACL
“Nebenwirkung”: Rechte und Rollen wirken auch lokal
 - Maximaler Internetzugriff
schneller “Generalschalter” für eine Datenbank



Dokument

- Leser-Felder
 - wenn vorhanden und nicht leer: exklusiv
 - bei mehreren: additiv
- Bearbeiter-Felder
 - zählen auch als Leser-Felder
 - relevant nur für Autoren
 - mehrere wirken additiv
- Dokument-Eigenschaften
 - auch für normale Nutzer
 - getrennt für jedes Dokument
 - gespeichert in \$Readers-Item





Kontrollierter Zugriff-Abschnitt

- nur von Entwicklern nutzbar
- beschränken Bearbeiter-Zugriff für Felder eines Abschnitts



Pragmatische Einführung in die Kryptographie



Symmetrische Verschlüsselung

- symmetrisch: gleicher Schlüssel zum Ver- und Entschlüsseln
- Algorithmen performant
- „Brute-Force-Angriffe“ (alle Schlüssel ausprobieren) auch!
- Schlüssellänge bestimmt Sicherheit: länger ist besser ;-)
- Problem: Wie verteilt man den Schlüssel?
- man braucht einen „sicheren Kanal“



Asymmetrische Verschlüsselung

- asymmetrisch: zwei verschiedene Schlüssel zum Ver- und Entschlüsseln
- relativ langsam (im Vergleich zu symmetrischen Verfahren)
- kürzere Schlüssel für vergleichbare Sicherheit
- zwei Schlüssel:
 - privater Schlüssel: ist geheim, darf nur der Besitzer kennen
 - öffentlicher Schlüssel: darf und soll (!) öffentlich bekannt sein
- Verschlüsselung mit dem öffentlichen Schlüssel: kann also jeder
- Entschlüsselung mit dem privaten Schlüssel: kann nur sein Besitzer



Asymmetrische Verschlüsselung (forts.)

- „Falltür“-Funktion:
 - in die eine Richtung geht es einfach...
 - aber man kann mit dem verschlüsselten Text (Chiffre) und dem öffentlichen Schlüssel nicht auf die ursprüngliche Information zurück kommen.
- Asymmetrische Verfahren sind auch ein bisschen symmetrisch: man kann mit privatem Schlüssel verschlüsseln und dann mit öffentlichen Schlüssel entschlüsseln



Telefonbuch für öffentliche Schlüssel

- öffentlicher Schlüssel wird in Verzeichnissen bekannt gemacht
- gebraucht wird eine Art „Telefonbuch“ mit Name —> Schlüssel-Einträgen
- Domino-Verzeichnis ist so ein Verzeichnis:

Person: **Thomas Bahn/assono** `tbahn@assono.de`

Basics | Work/Home | Other | Miscellaneous | **Certificates** | Roaming | Administration

Notes Certificates | Internet Certificates | Flat Name Key

Notes Certificate(s)

Notes certificate:	Present
Notes certified public key:	03002E02 A040F2C8 08G01617 G0020654 C5C27C03 G0030200 01208600 63CF2000 156F25G0 024FG002 63CF2000 146F2500 6C732000 597D25C1 01A07700 63CF2000 156F25G0 024FG002 63CF2000 146F2500 6C732000 597D25C1 4F3D6173 736F6E6F



Trick 17

- Symmetrische Verschlüsselung ist schnell, hat aber das Problem der Schlüsselverteilung.
- Asymmetrische Verschlüsselung hat kein Problem bei den Schlüsseln, ist aber deutlich langsamer (bei gleicher Sicherheit)
- Ein häufiger Trick besteht darin, beides zu kombinieren:
 - Es wird zufällig ein langer Schlüssel für die symmetrische Verschlüsselung vom Sender erzeugt.
 - Dieser wird asymmetrisch verschlüsselt mit dem öffentlichen Schlüssel des Empfängers und an diesen gesendet.
 - Er (und nur er) kann ihn entschlüsseln. So teilen beide Seiten den gleichen geheimen Schlüssel.
 - Die restliche Kommunikation wird symmetrisch mit diesem Schlüssel verschlüsselt.



Signaturen

- Verschlüsseln: nur bestimmte Personen dürfen Information sehen
- Signaturen: Beweis, dass Information
 1. wirklich von bestimmter Person stammt und
 2. nicht verändert wurde
- (kryptografische) Hash-Funktion erzeugt Prüfsumme:
 - macht aus einem langen Text eine Zahl fester Länge
 - ergibt eine ganz andere Zahl, wenn der Text nur ganz wenig verändert wird
- Signieren: Hash-Wert des Textes mit privaten Schlüssel verschlüsseln (kann also nur der Besitzer des privaten Schlüssels)



Signaturen (forts.)

- Prüfen: verschlüsselten Hash-Wert mit öffentlichen Schlüssel entschlüsseln und mit selbst berechneten Hash-Wert vergleichen
- Nur der Besitzer des privaten Schlüssels kann die Prüfsumme so verschlüsseln, dass er mit dem öffentlichen Schlüssel entschlüsselt werden kann.
- Wurde der Text nach der Signatur verändert, verändert sich auch sein Hash-Wert, so dass beim Vergleich die Änderung entdeckt wird.



Zertifikate und ID-Dateien



Zertifikate

- „Ein Zertifikat ist ein elektronischer Stempel zur Identifizierung eines Benutzers oder Servers.“ [Ebel2004], S. 322
- technischer: Ein Zertifikat ist im Wesentlichen die Signatur von Benutzerinformationen.
- Zertifikate...
 - werden von zentraler Stelle, der Certification Authority (CA = Zertifizierungsstelle), ausgestellt und können mit ihrem öffentlichen Schlüssel geprüft werden,
 - sind normalerweise zeitlich nur beschränkt gültig,
 - beweisen, dass die Benutzerinformationen „echt“ und unverändert sind.



Zertifizierer-ID

- Bei Notes/Domino: Bei der Installation des ersten Servers einer Notes-Domäne wird eine zentrale Zertifizierungsstelle, die Zertifizierer-ID erstellt.
- Zertifizierer-ID enthält privaten und öffentlichen Schlüssel. Ihr öffentlicher Schlüssel steht auch im Domino-Verzeichnis.
- OU-Zertifizierer funktionieren analog.



Registrierung neuer Benutzer & Server

- Beim Erstellen wird für alle Benutzer und Server beim Erstellen ein Zertifikat von der benutzten (OU-)Zertifizierer-ID ausgestellt und in der ID-Datei und dem Domino-Verzeichnis (Personen-/Server-Dokument) gespeichert.
- Sie können damit beweisen, dass sie wirklich mit der Zertifizierer-ID erstellt wurden!
- Andere Benutzer/Server können die Informationen im Domino-Verzeichnis nutzen, um die Echtheit des vorgezeigten Zertifikats zu prüfen (Authentifizierung) und dem Benutzer/Server vertrauen.



CA-Prozess

- CA-Prozess entkoppelt Erstellung der Benutzer von Signatur (und speichert die Zertifikate zusätzlich zur ID in admin4.nsf statt certlog.nsf)
- ohne CA-Prozess: Zugriff auf Zertifizierer-ID nötig für neue Benutzer und Rezertifizierungen
- eingeschränkte Rechte für Helpdesk-Mitarbeiter
- für ADSync und andere Tools, die neue Notes-Benutzer automatisiert erzeugen sollen



Notes-Gegenzertifikate

- Innerhalb einer Organisation (Notes-Domäne) gibt es das gemeinsame Domino-Verzeichnis, was ist aber mit fremden Notes-Benutzern und Domino-Servern?
- Dafür gibt es Notes-Gegenzertifikate!
- Der Name und der öffentliche Schlüssel eines fremden Zertifizierers, Servers oder Benutzers werden mit einer eigenen (OU-)Zertifizierer-ID (oder Server-ID) signiert und in das Domino-Verzeichnis eingetragen.
- Bei der Authentifizierung fremder Benutzer und Server werden dann die (überprüfbaren) Informationen aus dem Gegenzertifikat-Dokument im Domino-Verzeichnis verwendet.



ID-Dateien

- ID-Dateien eines Benutzers enthalten (u. a.)
 - Namen des Besitzers
 - Zertifikat einer (OU-)Zertifizierungs-ID
 - öffentlichen Schlüssel
 - privaten Schlüssel
 - ggf. Internet-Zertifikate (für SSL und S/MIME)
 - ggf. geheime Verschlüsselungsschlüssel (heißen nun mal so ;-)
- Aus dem vergebenen Kennwort wird ein Schlüssel berechnet, mit dem die privaten Daten in der ID-Datei symmetrisch verschlüsselt gespeichert werden. So kann man selbst wenn man die ID-Datei hat ohne das Kennwort nicht an diese Informationen kommen!



Vergessene Passwörter und verlorene ID-Dateien

- ID-Dateien mit bekannten Passwörtern im Dateisystem aufbewahren (nicht machen!)
- Passwortwiederherstellung und Mail-in-DB für ID-Dateien (besser)
- ID Vault (am besten)
- Drittherstellerprodukte



Passwort-Wiederherstellung

- In ID-Dateien können Wiederherstellungsinformationen gespeichert werden, mit deren Hilfe die privaten Angaben aus der ID entschlüsselt werden können.
- Diese Informationen werden verschlüsselt gespeichert, so dass normalerweise nur mehrere Administratoren zusammen die ID wiederherstellen können.
- Backups von den ID-Dateien werden dann – ebenfalls verschlüsselt – an eine bestimmte Mail- oder Mail-In-Datenbank geschickt. Diese Backups können im Fall des Verlusts oder der Beschädigung der ID-Datei verwendet werden, um eine neue ID-Datei für den Benutzer zu erstellen.



Verschlüsselung – vom Netzwerk bis zum Feld



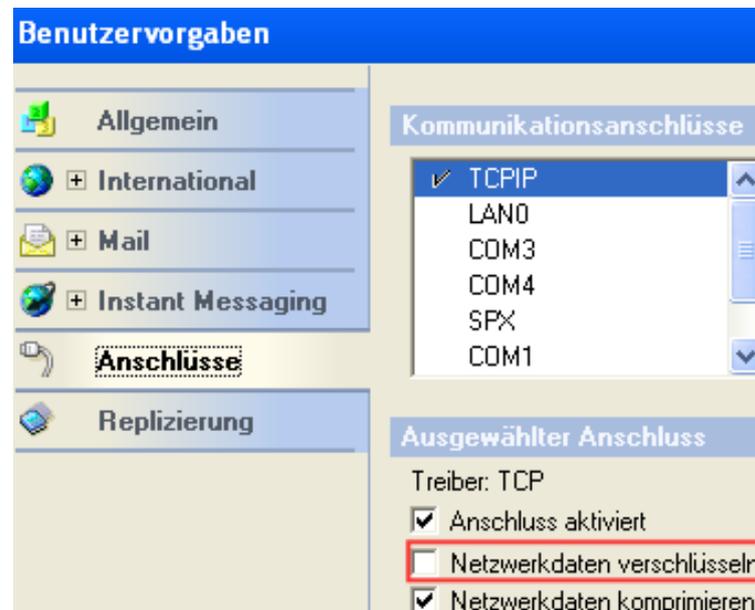
Verschlüsselung

- In Notes/Domino gibt es viele Ebenen, auf denen verschlüsselt werden kann:
 - Netzwerk
 - Datenbanken
 - nlo-Dateien (DAOS)
 - ein- und ausgehende E-Mails
 - Felder in Dokumenten



Netzwerkverschlüsselung

- Der Netzwerk-Verkehr kann bei Notes/Domino verschlüsselt werden.
- Wenn mindestens eine Seite – der Client oder der Server – verschlüsseln möchte, wird verschlüsselt.
- Verschlüsselung wird je Port konfiguriert:
- Client:





Netzwerkverschlüsselung (forts.)

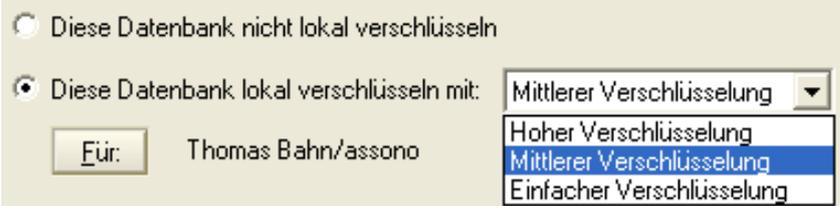
- Server:

The screenshot shows the Domino administration console interface. The main window displays the configuration for a server named "Domino-003/web.assono". A context menu is open over the "Anschlüsse" (Connections) section, with the "Konfiguration..." option highlighted. A dialog box titled "Anschlusskonfiguration für Domino-003" is open, showing the configuration for the selected "TCPIP" connection. The "Ausgewählter Anschluss" (Selected Connection) is "TCPIP". The "Treiber" (Driver) is "TCP". The "Anschluss aktiviert" (Connection active) checkbox is checked. The "Netzwerkdaten verschlüsseln" (Encrypt network data) checkbox is checked and highlighted with a red box. The "Netzwerkdaten komprimieren" (Compress network data) checkbox is unchecked.



Verschlüsselung von Datenbanken

- Notes-Datenbanken und -Schablonen können verschlüsselt auf dem Datenträger gespeichert werden.
- Bei Server wird mit dessen öffentlichen Schlüssel verschlüsselt, beim Notes-Client mit dem öffentlichen Schlüssel aus der ID-Datei des aktuellen Benutzers. So kann nur der Server bzw. Benutzer die Datenbank mit dem jeweiligen privaten Schlüssel entschlüsseln.
- Es gibt drei Stufen:
- Umso höher die Verschlüsselung, desto sicherer ist sie, aber auch langsamer
- Entweder beim Anlegen der Datenbank oder einer Replik gleich richtig einstellen oder nach einer Änderung die Datenbank komprimieren (compact)
- sinnvoll nur, wenn Server-ID gesichert



The screenshot shows a dialog box for database encryption settings. It has two radio buttons: 'Diese Datenbank nicht lokal verschlüsseln' (unselected) and 'Diese Datenbank lokal verschlüsseln mit:' (selected). Below the second radio button is a text field containing 'Für: Thomas Bahn/assono'. To the right of the text field is a dropdown menu with three options: 'Mittlerer Verschlüsselung' (selected), 'Hoher Verschlüsselung', and 'Einfacher Verschlüsselung'.



nlo-Dateien (DAOS)

- Vorgabe: nlo-Dateien werden mit privaten Schlüssel aus Server-ID verschlüsselt
- sinnvoll überhaupt nur, wenn Server-ID gesichert
- macht aber einiges umständlicher
 - Rücksicherung auf anderen Server
 - Deduplizierung auf SAN
 - Server-Starts
- Ausschalter: `DAOS_Encrypt_NLO=0` in `notes.ini`
- muss konfiguriert werden, bevor DAOS auf Server aktiviert wird



E-Mails (Notes-intern)

- Ein- und ausgehende E-Mails können verschlüsselt werden.
- Die Verschlüsselung eingehender E-Mails wird im Personen-Dokument konfiguriert:

Person: **Thomas Bahn/assono** `tbahn@assono.de`

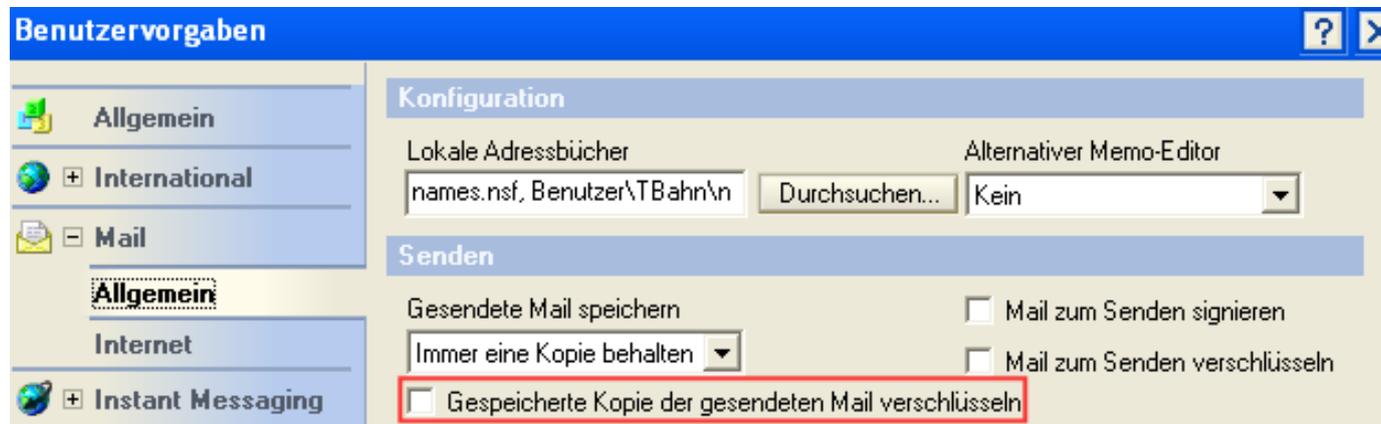
Basics | Work/Home | Other | Miscellaneous | Certificates | Roaming | Administration

Basics		Mail	
First name:	Thomas	Mail system:	Notes
Middle name:		Domain:	assono
Last name:	Bahn	Mail server:	Domino-001/assono
User name:	Thomas Bahn/assono Thomas Bahn	Mail file:	mail\tbahn
		Forwarding address:	
		Internet address:	tbahn@assono.de
		Format preference for incoming mail:	Keep in senders' format
		When receiving unencrypted mail, encrypt before storing in your mailfile:	No



E-Mails (Notes-intern; forts.)

- Beim E-Mail-Versand muss man noch unterscheiden:
 1. Die Verschlüsselung einer ggf. beim Absender gespeicherten Kopie der E-Mail wird in den Benutzervorgaben eingestellt:





E-Mails (Notes-intern; forts.)

2. Die Verschlüsselung versendeter E-Mails kann der Benutzer selbst bestimmen, entweder in den Zustelloptionen oder direkt in der Maske unter der Aktionsleiste:

Zustelloptionen

Allgemein | Erweitert

Zustelloptionen

Dringlichkeit: Normal

Zustellungsbericht: Nur bei Fehler

Zustellungspriorität: Normal

Keine Benachrichtigung an mich senden, wenn Empfänger...

Sicherheitsoptionen | Modus

Signieren

Verschlüsseln

Diese Sicherheitsoptionen als Vorgabe speichern

Sichern | Adresse... | Zustelloptionen... | Werkzeuge ▾

Signieren | Verschlüsseln



Verschlüsselung von Feldern in Dokumenten

- Entwickler können für jedes Feld einer Maske einstellen, dass es verschlüsselt werden soll: Feld-Eigenschaften – Erweitert – Sicherheitsoptionen auf „Verschlüsselung für dieses Feld aktivieren“
- Bei RichText-Feldern werden auch die angehängten Dateien verschlüsselt gespeichert. Bei Kennwort-Feldern wird die Option automatisch gesetzt.
- Zusätzlich muss ein Schlüssel festgelegt werden (oder mehrere).
- Dafür gibt zwei Möglichkeiten:
 - öffentliche Schlüssel von Benutzern
 - geheime Verschlüsselungsschlüssel



Verschlüsselung von Feldern in Dokumenten (forts.)

1. öffentliche Schlüssel von Benutzern

Es muss ein Feld `PublicEncryptionKeys` geben, in das die `NotesNamen` der Personen eingetragen werden müssen, für die das Dokument lesbar sein soll.

Beim Speichern oder Senden des Dokuments werden über die `NotesNamen` die Personen-Dokumente im Domino-Verzeichnis gesucht und der öffentliche Schlüssel der Benutzer ermittelt.

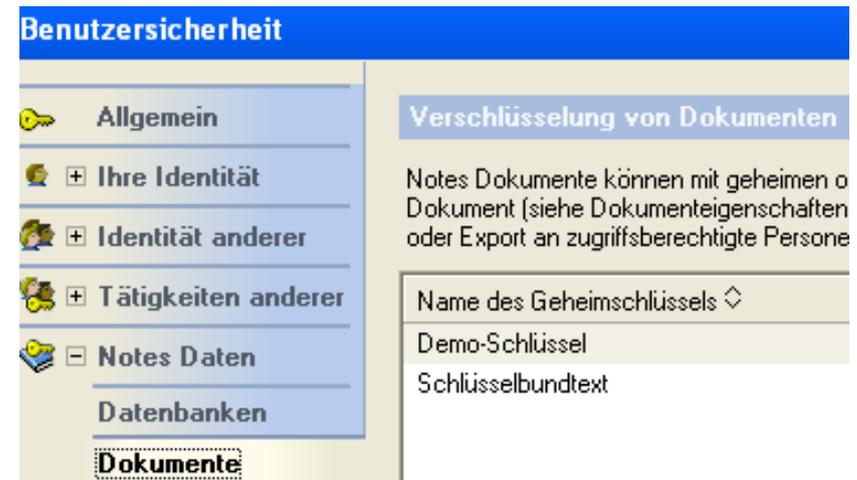
Dann werden die gekennzeichneten Felder gegen diese Schlüssel verschlüsselt.



Verschlüsselung von Feldern in Dokumenten (forts.)

2. geheime Verschlüsselungsschlüssel

- Über die Sicherheitsoptionen – Notes-Daten – Dokumente können sog. Geheimschlüssel erstellt werden.
- Diese werden in der ID-Datei gespeichert.
- Sie können exportiert oder per E-Mail versendet und dann in andere ID-Dateien importiert werden.

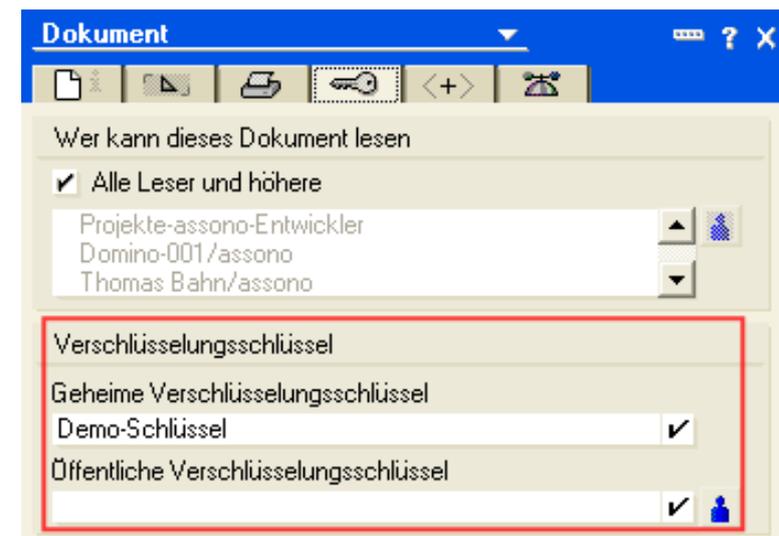




Verschlüsselung von Feldern in Dokumenten (forts.)

2. geheime Verschlüsselungsschlüssel (forts.)

- In den Masken-Eigenschaften kann ein Entwickler unter Sicherheit einen Standard-Verschlüsselungsschlüssel aus den Geheimschlüsseln in seiner ID-Datei auswählen.
- Enthält ein Dokument ein Feld SecretEncryptionKeys, dann erwartet Notes darin den oder die Namen von geheimen Verschlüsselungsschlüsseln.
- In den Dokument-Eigenschaften – Sicherheit kann man für jedes Dokument einzeln geheime oder öffentliche Verschlüsselungsschlüssel (in Form von Personen) auswählen.





Signaturen



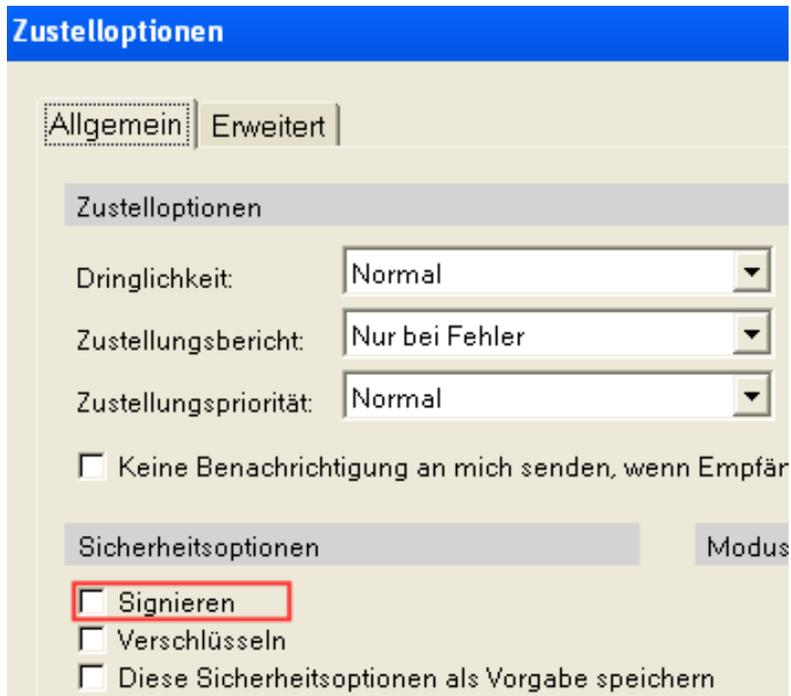
Signaturen

- Auch Signaturen gibt es bei Notes/Domino auf verschiedenen Ebenen:
 - ausgehende E-Mails
 - Dokumente
 - Abschnitte



Ausgehende E-Mails (Notes-intern)

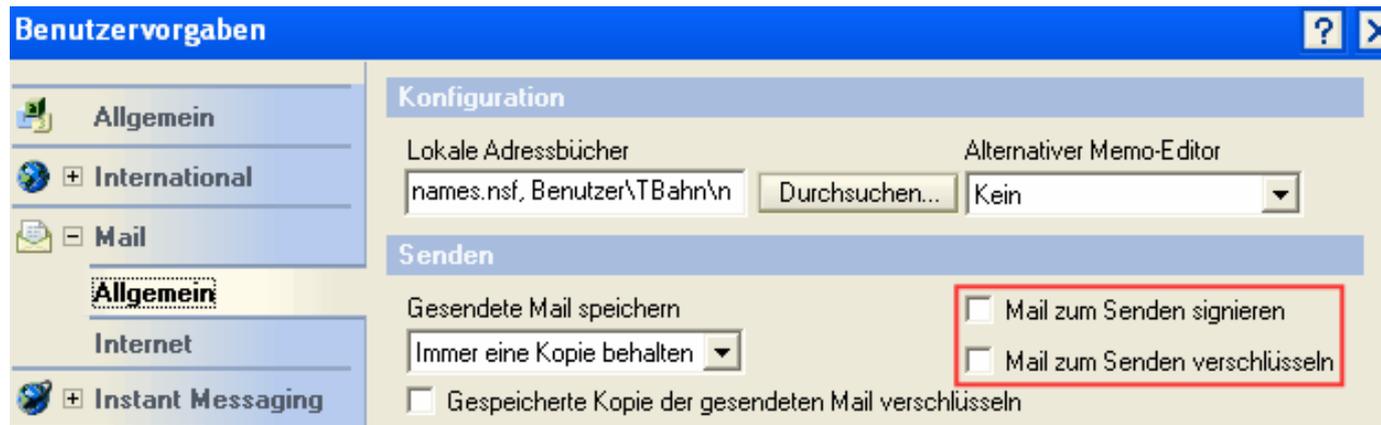
- Der Absender kann wie bei der Verschlüsselung selbst bestimmen, ob eine E-Mail signiert werden soll. Dies passiert wieder entweder in den Zustelloptionen oder direkt in der Maske unter der Aktionsleiste:





Ausgehende E-Mails (Notes-intern)

- Und wie bei der Verschlüsselung kann man in den Benutzeroptionen die Vorgabe für das Signieren einstellen:





Dokumente

- Ein Entwickler kann in den Masken-Eigenschaften einstellen, dass alle Dokumente, die mit dieser Maske gespeichert oder versendet werden, automatisch mit der aktuellen Benutzer-ID signiert werden:

Optionen	<input type="checkbox"/>	Felder automatisch aktualisieren		
	<input type="checkbox"/>	Anonyme Maske		
	<input type="checkbox"/>	Kein anfängl. Fokus	<input type="checkbox"/>	Kein Fokus bei F6
	<input checked="" type="checkbox"/>	Dokumente signieren, die diese Maske verwenden		
	<input type="checkbox"/>	In Notes als HTML rendern		

- Wird so ein Dokument geöffnet, stehen in der Statuszeile die Details der Signatur:

Signiert durch Thomas Bahn/assono am 10.09.2007 01:23:36, gemäß /assono



Abschnitte

- Zusätzlich können auch Kontrollierter-Zugriff-Abschnitte signiert werden. Das sieht dann zum Beispiel so aus:

▼ **Stellungnahme Vorgesetzter - Signiert durch Thomas Bahn/assono am 10.09.2007 01:44:02, gemäß /assono**

Stellungnahme

Maßnahme wie vorgeschlagen durchführen.

- Dazu muss der Entwickler in den Feldeigenschaften – Erweitert – Sicherheitsoptionen „Signieren beim Versenden oder Speichern im Abschnitt“ für mindestens ein Feld innerhalb des Abschnitts auswählen.
- Beim Speichern oder Senden eines Dokuments wird dann die Signatur jedes Abschnitts aktualisiert, den der aktuelle Benutzer bearbeiten darf.



Authentifizierung



Authentifizierung vs. Autorisierung

- Authentifizierung: Wer bin ich?
- Autorisierung: Was darf ich?



Anmeldeprozess

- Anmeldung des Notes-Clients beim Domino-Server
- Ist der Benutzer wirklich der, der er behauptet zu sein?
- Ist der Server wirklich der, der er behauptet zu sein?
- gilt auch zwischen zwei Servern



Anmeldung am Server

- Bei der Anmeldung am Server werden 2 Prüfungen vorgenommen:
- Validierung des öffentlichen Schlüssels: mit Hilfe des Zertifikats wird der öffentliche Schlüssel aus der ID-Datei geprüft
- gegenseitige Authentifizierung mit Challenge/Response-Verfahren:
 1. Server erzeugt Zufallszahl, verschlüsselt sie mit öffentlichem Schlüssel des Benutzers und überträgt das Ergebnis
 2. Benutzer entschlüsselt die Zahl und überträgt sie mit dem öffentlichen Schlüssel des Servers verschlüsselt zurück
 3. Der Server entschlüsselt sie und vergleicht sie mit der ursprünglichen Zufallszahl: Stimmen die Zahlen überein, muss es der richtige Benutzer sein.
- und danach noch einmal anders herum



Passwort-Management



Sicherheitseinstellungen (Richtlinien)

- Kennwortverwaltung – Allgemein – Kennwortverwaltungsoptionen
- Benutzerdefinierte Kennwortrichtlinie für Notes-Clients verwenden: detaillierte Vorgaben für das Kennwort
- Kennwort überprüfen anhand der Notes-ID-Datei: Ja
 - Problem: ID gestohlen oder kompromittiert
 - Passwort nur auf ID-Datei – Passwortänderung wirkt nicht auf alte ID-Datei
 - Überprüfung anhand Digest in Personen-Dokument
- Internetkennwort bei Änderung des Notes-Client-Kennworts aktualisieren: Ja
- Andere Notes-basierte Programme fragen kein Kennwort ab (vermindert Sicherheit): Nein



Sicherheitseinstellungen (Richtlinien)

- Kennwortverwaltung – Allgemein
- Einmalige Anmeldung von Windows für Standard-Notes-Client aktivieren:
 - "SPNEGO", gleich mehr
- Sperreinstellungen für Internetkennwörter
- Sperreinstellungen für Internetkennwörter des Servers überschreiben?
- Anmerkung: Der Server muss die Sperre für Internetkennwörter erzwingen, damit diese Richtlinieneinstellungen wirksam werden.



Gemeinsame Notes-Anmeldung

- Wenn die gemeinsame Notes-Anmeldung aktiviert ist, haben Notes-IDs keine Notes-Kennwörter mehr. Stattdessen wird zum Schutz der ID ein komplexes "Secret" verwendet. Dieses "Secret" wird anhand eines Microsoft Windows-Sicherheitsmechanismus verschlüsselt und auf den Computern der Benutzer lokal gespeichert.



Integrated Windows Authentication (IWA)

- für Eclipse-basierte Clients
- Integrated Windows Authentication (IWA) steht für bereitgestellte und Eclipse-basierte Clientanwendungen von Drittanbietern zur Verfügung und ermöglicht die SPNEGO-Authentifizierung für Eclipse-basierte Funktionen und Anwendungen im Notes-Client. Dies schließt Widgets und Live Text, Feeds, IBM Connections und Verbundanwendungen sowie integriertes IBM Sametime und Symphony ein. IWA funktioniert auch mit Produkten, die auf Eclipse basieren, aber nicht in Notes integriert sind, beispielsweise IBM WebSphere Portal mit SiteMinder und eigenständigem Connections 3.0 mit SiteMinder.
- Anmerkung: IWA kann nicht als Mechanismus zur Authentifizierung beim Start des Notes-Clients verwendet werden.



Föderierte Anmeldung

- Durch die Authentifizierung mittels föderierter Identität mit dem Security Assertion Markup Language-Standard (SAML) entfällt für Notes-Client-Benutzer die Eingabe eines Notes-Kennworts über die föderierte Notes-Anmeldung. Die Benutzer-IDs müssen in einer ID-Vault gespeichert werden, deren Domino-Server mit Hostnamen für Identitäts-Provider-Partnerschaften (IdP-Partnerschaften) konfiguriert ist. Der Inhalt der ID-Datei der Notes-Client-Benutzer wird nach dem Download aus der ID-Vault im Speicher auf dem Client abgelegt.



Sicherheit in Notes-Anwendungen



Echte Sicherheit vs. gefühlte Sicherheit

- „Echte Sicherheit“
 - Verschlüsselung!
 - Leser-/Bearbeiter-Felder
 - Zugriffskontrolle (ACL)
- „Gefühlte Sicherheit“
 - Hide-When-Formeln
 - Berechtigungen auf Ansichten, Selektionsformeln
 - versteckte Gestaltung
 - Selektive Replikation
 - Öffnen im Bearbeiten-Modus verhindern
 - Keine bearbeitbaren Felder in Maske



Datenbank-Eigenschaften

- Exportieren von Ansichtsdaten verhindern
- gilt auch für Kopieren als Tabelle
- verhindert nicht Drucken oder Kopieren von Dokumentinhalten



Ansichten-Eigenschaften

- Wer darf diese Ansicht verwenden?
- Verfügbar für Benutzer mit öffentlichem Zugriff



Masken-Eigenschaften

- Dokumente signieren, die diese Maske verwenden; \$Signature; \$SignatureStatus (geöffnetes Dokument)
- Standard-Verschlüsselungsschlüssel; SecretEncryptionKeys; \$EncryptionStatus (geöffnetes Dokument)
- Standard-Lesezugriff für mit dieser Maske erstellte Dokumente; \$Readers
- Wer kann mit dieser Maske Dokumente erstellen; aber: Kopieren und Einfügen, programmatisch per Agent erstellen, deshalb ACL!
- Drucken/Weiterleiten/Kopieren verhindern: \$KeepPrivate = "1"
- Verfügbar für Benutzer mit öffentlichem Zugriff (Maske)
- \$PublicAccess = "1" (Dokument)



Feld-Eigenschaften

- Verschlüsselung für dieses Feld aktivieren; \$SealData
- Signieren beim Versenden oder Speichern im Abschnitt
- Mindestens Editorzugriff für Bearbeitung erforderlich
- Flags des Items: SIGN SEAL PROTECTED
- Eingabe-Formel



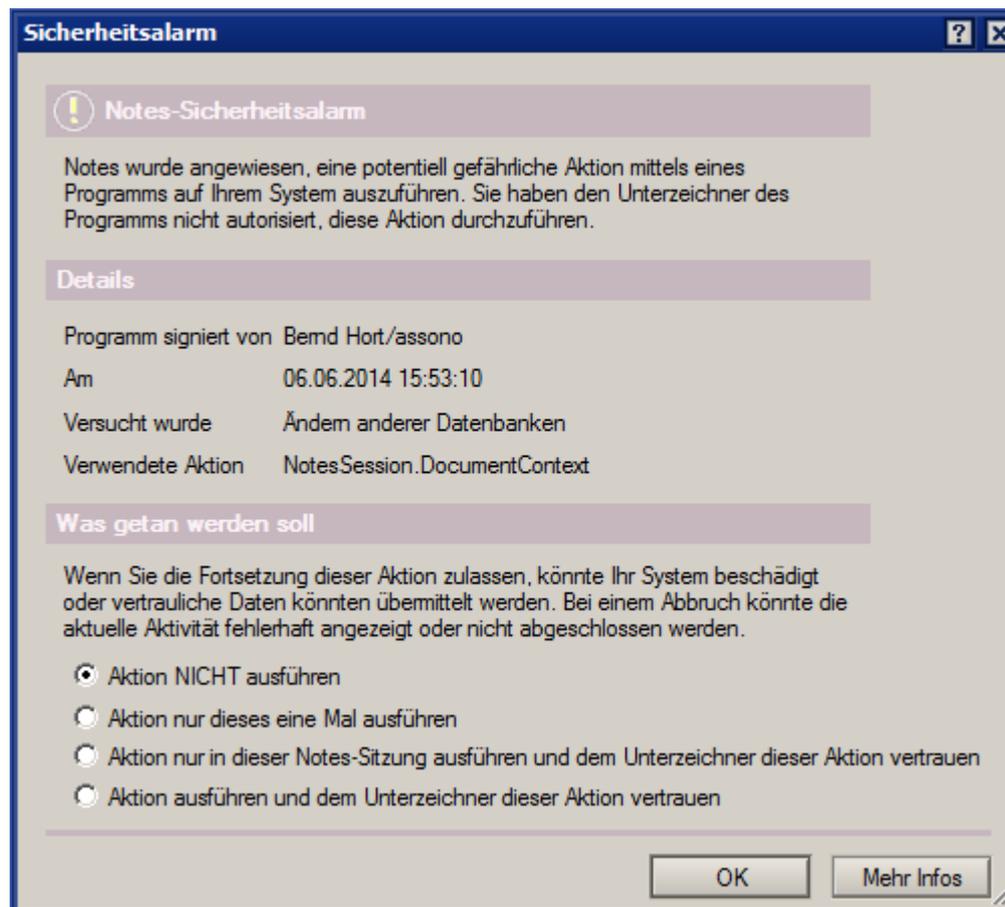
Kontrollierter Zugriff-Abschnitt

- Wer darf bearbeiten: Liste oder Formel
- Signieren des Abschnitt-Inhalts
- Item `_unbenannt_`: Editoren
- Item `$Sig__unbenannt_`: Signatur



Execution Control List (ECL)

- Wer darf „gefährlichen“ Code ausführen
- Code ist immer signiert





Execution Control List (ECL; forts.)

- Datei – Sicherheit – Benutzersicherheit...

Benutzersicherheit

Einschränkungen für ausgeführte Programme, Agenten, Makros und Befehle

Notes führt 'Code' (Agenten usw.) aus, wenn Sie eine Notes-DB verwenden. Code kann signiert sein, um den Autor zu identifizieren. Darauf basierend können Sie die Möglichkeiten des Codes einschränken.

Wählen Sie einen Namen in der Liste, um dessen Berechtigungen zu sehen bzw. zu setzen.
-Default- gilt, wenn der Code von einer nicht aufgelisteten Person signiert ist.
-Keine Signatur- gilt, wenn der Code-Autor unbekannt ist.

Wenn Code signiert ist von:

- Lotus Notes Template Development/Lotus N
- Thomas Bahn/assono

Zugriff zulassen auf:

- Dateisystem
- Externen Code
- Aktuelle Notes-Datenbank
- Umgebungsvariablen
- Netzwerk
- Externe Programme
- Nicht-Notes-Datenbanken

Folgende Funktionen zulassen:

- Mail senden
- Andere Notes-DBs lesen
- Vom Eigenschaftsbroker lesen
- Ändern Ihrer Ausführungskontrollliste
- Widget-Funktionen konfigurieren
- Java-Code laden
- Daten exportieren
- Andere Notes-DBs ändern
- In Eigenschaftsbroker schreiben

Hinzufügen... Umbenennen... Entfernen

Effektive Sitzungs-ECL

Alle aktualisieren

Workstation-, Applet- und JS-ELCs zuletzt geändert: 29.09.2014 10:08:53

OK Schließen



Execution Control List (ECL; forts.)

- per Sicherheitseinstellung und Richtlinie administrativ steuerbar

Sicherheitseinstellungen : Standard-Security-Settings

Allgemein | Kennwortverwaltung | **Ausführungskontrollliste (ECL)** | Schlüssel und

ECL

Administrations-ECL: -Default-

Aktualisierungsmodus: ▾

Aktualisierungsintervall: ▾



Fragen?

jetzt stellen – oder später:

 tbahn@assono.de

 <http://www.assono.de/blog>

 04307/900-401



Folien unter:

[www.assono.de/blog/d6plinks/
AC14-Weil-sicher-sicher-sicher-ist](http://www.assono.de/blog/d6plinks/AC14-Weil-sicher-sicher-sicher-ist)