Comtarsia SignOn Proxy 2008

Technische Funktionsbeschreibung

Build 5.1.3.51
21.02.2011

Inhaltsverzeichnis

1. Einleitung

2. Grundlegende Konfiguration

[HKLM\SOFTWARE\Comtarsia\ SignOn Solutions 2008]

REG_SZ:"path"="%ProgramFiles%\\Comtarsia\\SignOn Solutions 2008"

Der Installationspfad.

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\ SignOnProxy]

REG_DWORD:"listenerPort"=0x000007d3

Definiert den TCP-ListenerPort des SignOn Proxy. Default ist 2003.

REG_SZ:"listenerInterface"="*"

Definiert das Listener Interface. Bei „*“ oder „“ binded sich der SignOn Proxy auf alle verfügbaren Interfaces.

Beispiel: "listenerInterface"="192.168.1.1"

REG_DWORD:"rcvTimeout"=0x00000004

Das Socket Receive Timeout in Sekunden. Diese Zeit gilt für das Empfangen des Client-Pakets. Anschliessend wird die TTL des Clients verwendet.

REG_DWORD:"connectTimeout"=0x00000005

Definiert das maximale Socket Connect Timeout für Verbindungen zum SignOn Agent in Sekunden. Ist die TTL des Client-Pakets kleiner als das connectTimeout so wird die TTL verwendet.

REG_SZ:"tlsCADir"=""

Definiert den Pfad zu einem absoluten Verzeichnis, in welchem sich ein oderer mehrere CA-Zertifikate befinden. Nach jeder Änderung in diesem Verzeichnis muss OpenSSL-„c_rehash“ aufgerufen werden.

Es darf nur entweder CADir oder CAFile gesetzt werden.

REG_SZ:"tlsCAFile"="%ProgramFiles%\\Comtarsia\\SignOn Solutions 2008\\cert\\ca.pem"

Definiert einen absoluten Pfad zu einer Datei, welche ein oder mehrere CA-Zertificate enthält. Sollen mehrere Zertifikate in dieser Datei gespeichert werden, so werden siese einfach hintereinander kopiert.

Es darf nur entweder CADir oder CAFile gesetzt werden.

REG_SZ:"tlsKeyFile"="%ProgramFiles%\\Comtarsia\\SignOn Solutions 2008\\cert\\proxy.pem"

Ein absoluter Pfad zu einer Private Key-Datei, welche für die Kommunikation mit den anderen SignOn Gate-Komponenten verwendet wird.

REG_SZ:"tlsCertFile"="%ProgramFiles%\\Comtarsia\\SignOn Solutions 2008\\cert\\proxy.pem"

Ein absoluter Pfad zu einem Zertifikat, welches für die Kommunikation mit den anderen SignOn Gate-Komponenten verwendet wird.

REG_DWORD:"tlsOptions"=0x00000000

Wird derzeit nicht verwendet.

REG_DWORD:"trustOptionsClient"=0x00000000

Definiert die SSL Verifizierungsoptionen fuer die Kommunikation zwischen den Clients und dem SignOn Proxy.

Flag	Bezeichnung	Beschreibung
0	No Check	Keine Überprüfung
1	Accept List	Überprüfung mittels Accept List (Liste von IP-Adressen)
2	Certificate OIDs	Es wird auf das Vorhandensein der Comtarsia OIDs geprüft. Für Details hierzu siehe das Dokument „SignOnGate Certificates“.
0x100	Certificate FQDN	Es wird der FQDN des Zertifikates überprüft. Dieser FQDN muss mit dem vom DNS geliefertetem Reverse Lookup-Ergebnis uebereinstimmen.

REG_DWORD:"trustOptionsServer"=0x00000000

Definiert die SSL Verifizierungsoptionen für die Kommunikation zwischen dem SignOn Proxy und den SignOn Agents. Beschreibung der Flags siehe oben.

REG_DWORD:"clientAcceptFilter"=0xffffffff

Definiert einen Filter über die Client-Typen, welche sich zum SignOn Proxy verbinden können.

Flag	Client-Typ
0x02	Logon Client
0x04	Web Client/Web Gateway
0x08	Security Agent
0x10	SignOn Proxy
0x20	LDAP Directory Replicator

REG_DWORD:"nrOfWorkerThreads"=0x4

Definiert die Anzahl der Client-Bearbeitungs-Threads

REG_DWORD:"authenticationMode"=0x00000001

Definiert den Authentifizierungsmodus: 0 = kein Gegencheck, 1 = LDAP Gegencheck

REG_DWORD:"syncClientLogonDC"=0x00000000

Ist diese Funktion aktiv, so versucht der Proxy nach Moeglichkeit, den Synchronisationsrequest des Benutzers nach Moeglichkeit auf den vom Client ausgewählten AD Logon Server. Voraussetzung hierfuer ist, dass auf diesem Server ein SignOn Agent installiert ist und dieser auch als „SyncDomain“-Server konfiguriert ist.

REG_DWORD:"agentProcessTimeout"=0x000493E0

Definiert ein Timeout in Millisekunden, wann ein einmal nichtfunktionierender Agent wieder verwendet wird. Default sind 300000 ms (=5 Minuten)

REG_SZ:"profileName"=""

Der Name des Profils.

REG_SZ:"profileComment"=""

Ein Kommentar für das Profil.

3. LDAP Konfiguration

Sämtliche LDAP-Konfigurationswerte befinden sich unter den Registry-Key:

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\LDAP\Servers\<LDAP-Server>]

Über den Namen des Registry-Keys „<LDAP-Server>“ wird der Hostname/die IP des LDAP Servers angegeben.

3.1. Allgemeine Server Parameter

REG_SZ: "failoverHost"=""

Hier kann ein Failover LDAP Server eingetragen werden, welcher dann vom Proxy bei Nichterreichbarkeit des primären LDAP Servers (Communication error) kontaktiert wird.

REG_SZ:"baseDN"="o=comtarsia"

Die BaseDN des LDAP Servers

REG_DWORD:"timeout"=0x000000a

Definiert das LDAP-Timeout in Sekunden. Dieser Wert ist als Maximalwert pro LDAP-Aktion zu sehen, das komplette Timeout wird durch den jeweiligen Client vorgegeben

REG_DWORD:"port"=0x0000185

Der TCP-Port des LDAP-Servers. Default ist bei LDAP 389 und bei LDAP ueber SSL (LDAPS) 636.

REG_DWORD:"sslMode"=00000000

Mit diesem Parameter kann definiert werden, ob SSL verwendet werden soll und in welchem Modus.

Wert	**SSL Modus**
0	kein SSL
1	SSL ohne Client-Zertifikat, das Server-Zertifikat wird nicht überprüft
2	SSL ohne Client-Zertifikat, das Server-Zertifikat wird in diesem Modus überprüft.
3	SSL mit System-Zertifikat, das Server-Zertifikat wird überprüft und ein System-eigenes Zertifikat wird an den LDAP-Server zur überprüfung übermittelt. Das Zertifikat muss sich im Computer Certifikate store befinden und als CN den Hostname beinhalten.

Der SignOn Proxy verwendet die Microsoft Certificate Stores, d.h. das CA-Zertifikat und eventuell weitere Zertifikate in der Certificate Chain des LDAP-Servers müssen unter „Trusted Root Certificate Authorities“ des Service oder Computer-Accounts eingespielt werden.

REG_DWORD:"serverType"=0x0000000

Wert	**Server Typ**
0	Generisch
1	iPlanet
2	Netscape/Sun One DS
3	OpenLDAP
4	IBM DS RACF LDAP Gateway
5	IBM Lotus Domino
6	Novell eDirectory
7	IBM Direcory Server <5.1
8	IBM Direcory Server 5.1, 6.0

9	IBM Direcory Server >=6.1
10	Microsoft Active Directory
11	Fedora Directory Server
12	Sun DS Enterprise Edition DS 6.3

3.2. Benutzer Logon Modi

Der SignOn Proxy bietet verschiedene Möglichkeiten, die LDAP Benutzer-DN zu ermitteln.

REG_SZ:"userDNPrefix"="uid="

Das UserDNPrefix, Beschreibung siehe unten.

REG_SZ:"userDNSuffix"=",ou=users"

Das UserDNSuffix, Beschreibung siehe unten.

REG_DWORD:"appendBaseDN"=00000001

Definiert, ob die BaseDN automatisch an die Benutzer-DN angehängt werden soll oder nicht.

3.2.1. Static DN

In diesem Modus wird die Benutzer-DN immer gleich aussehen, es wird nur der Benutzername dynamisch eingesetzt. Die Benutzer-DN wird folgendermassen zusammengesetzt:

%USERDNPrefix%%USERNAME%%UserDNSuffix%,%BaseDN%

Beispiel:

userDNPrefix=“cn=“

userDNSuffix=“,ou=users“

baseDN=“o=comtarsia“

appendBaseDN=1

Ergibt für den Benutzernamen „User“ die folgende DN:

“cn=User,ou=users,o=comtarsia”

3.2.2. Search for User

In diesem Modus wird nach dem Benutzer im LDAP gesucht. Dadurch können die Benutzer im Gegensatz zum Modus „Static DN“ auch in verschiedenen OUs enthalten sein.

Die LDAP-Suche kann entweder anonym oder mit einem bestimmten Systembenutzer erfolgen.

REG_DWORD:"searchForUser"=00000000

Aktiviert oder deaktiviert den „Search for User“ Modus.

REG_DWORD:"ignoreNoUniqueUser"=00000000

Wenn bei der Suche nach dem Benutzer mehr als Ergebniss vom LDAP-Server zurückgeliefert wird, so wird dies per Default vom SignOn Proxy als Authentication-Error gewertet (=0). Bei Bedarf kann dies auch ignoriert werden (=1).

REG_DWORD:"failoverOnUserNotFound"=00000000

Hiermit kann definiert werden, ob der SignOn Proxy einen Failover auf den Failover-LDAP-Host machen soll, falls der Benutzer bei der Suche nicht gefunden wurde (=1).

3.2.3. OU Searchlist

Im Gegensatz zum “Search for User”-Modus, können hier eine oder mehrere OUs über ein LDAP-Attribut definiert werden, welche nach dem Benutzer durchsucht werden.

Detaillierte Beschreibung einer LDAP-Benutzeranmeldung mit der OUSearchList-Funktionalität:

1) Ist der OUSearchList-Modus aktiv, so verbindet sich der SignOn Proxy mit den in der Registry hinterlegten Credentials (siehe Kapitel 3.2.5 LDAP Systembenutzer) des LDAP-Servicebenutzers zum LDAP.

2) Wurden in der Registry ein LDAP-Objekt inkl. Attribut [3] hinterlegt, in welchem sich die OUSearchList befindet, so wird dieses ausgelesen.

3) Es wird nun für jeden Eintrag in der OUSearchList eine LDAP-Query abgesetzt, um zu sehen, ob der Benutzer in der jeweiligen OU existiert. Die Einträge der OUSearchList werden in der konfigurierten Reihenfolge durchgegangen. Wird der Benutzer in einer OU gefunden, so wird die weitere Suche abgebrochen. Wird der Benutzer in keiner der konfigurierten OUs gefunden, so wird der Logon abgebrochen und dem Benutzer der in „OUSearchListErrorCode“ konfigurierte Wert als Fehlermeldung angezeigt.

4) Der LDAP-Servicebenutzer wird wieder abgemeldet und der Logon-Benutzer mit der unter 3) ermittelten Benutzer-DN an das LDAP angemeldet. An dieser Stelle ist die neue OUSearchList Funktionalität beendet und die bestehende Logon Client/Proxy LDAP-Funktionalität wird weitergeführt.

REG_DWORD:"OUSearchListMode"=00000000

Mit diesem wert wird diese Funktion ein (=1) oder ausgeschltet (=0).

REG_DWORD:"OUSearchListErrorCode"=00000006

Definiert den Fehlercode, den der SignOn Proxy an den Client zurückliefert, falls der Benutzer nicht gefunden wurde.

Mögliche Werte sind:

Wert	Bedeutung
2	Falscher Benutzer und/oder Passwort. Wird dieser Wert gesetzt, so kann nicht mehr unterschieden werden, ob der Benutzer und/oder das Passwort falsch sind, was aus Sicherheitsgründen vorteilshaft ist.
6	Undefinierter Benutzer

REG_SZ:"OUSearchListObjectDN"=""

Definiert die LDAP-Objekt-DN, in welchem die OUSearchList hinterlegt ist. Die LDAPOUSearchListObjectDN muss eine absolute LDAP-DN sein.

REG_SZ:"OUSearchListAttribute"=""

Definiert das LDAP-Attribut, in welchem die OUSearchList hinterlegt ist. Das LDAPOUSearchListAttribute ist eine Single-Value String-Attribut, die einzelnen OUSearchList-Einträge werden mittels „;“ getrennt.

3.2.4. Smart Card Authentication

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\UserCertificateMapping]

Der Comtarsia SignOn Proxy bietet die Möglichkeit, eine direkte Smart Card Authentifizierung eines Benutzers durchzuführen. Dies erfolgt sozusagen stellvertretend für den LDAP-Server, wodurch der LDAP-Server selbst nicht für SSL Client Authentication unterstützen muss/konfiguriert sein muss.

Der Ablauf:

1) Als erstes wird das Zertifikat überprüft, ob es gültig ist und ob ein passendes CA-Zertifikat im lokalen Proxy-Store (siehe userCertificateCAFile/Dir) vorliegt.

2) Im nächsten Schritt wird dann eine vom Client mitgeschickte Signatur überprüft, wodurch der Proxy feststellen kann, ob der Benutzer auch wirklich einen Private Key zu dem präsentiertem Zertifikat hat.

3) Als nächstes wird nun das Certificate Mapping durchgeführt (siehe unten).

4) Nun verbindet sich der Proxy zum LDAP und sieht nach, ob der gemappte Smart Card Benutzer im LDAP existiert.

5) Je nach Konfiguration (siehe compareUserCertificate) wird jetzt noch das vom Benutzer präsentierte Zertifikat mit dem im LDAP-Benutzerobjekt abgelegten Zertifikaten verglichen.

REG_SZ:"userCertificateCAFile"=""

Definiert den Pfad zu einer Datei, welche ein oder mehrere CA-Zertifikate enthält.

REG_SZ:"userCertificateCADir"=""

Definiert den Pfad zu einem Verzeichnis, welches ein oder mehrere CA-Zertifikate enthalten kann. Nach jedem Update müssen neue Hashes für die Zertifikat berechnet werden, dies kann mittels des OpenSSL-Tools „c_rehash“ erfolgen.

REG_DWORD:"compareUserCertificate"=00000000

Mit diesem Parameter wird definiert, ob das Benutzerzertifikat mit dem im LDAP Benutzerobjekt hinterlegtem Zertifikaten verglichen werden soll (siehe oben, Ablauf 5). Die Zertifikate(Multi-Value Attribut) im LDAP-Benutzerobjekt können im PEM- oder DER-Format abgelegt sein. Gibt es bei keinem Zertifikat eine Übereinstimmung, so bricht der SignOn Proxy die Authentifizierung ab und es findet keine Benutzersynchronisierung statt. Gibt es mindestens eine Übereinstimmung, so wird der Authentifizierungs- und Synchronisationsprozess fortgesetzt.

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\UserCertificateMapping\<UserCertificateMappingName>]

<UserCertificateMappingName> muss in der Form „NNN_MAPPINGNAME“ sein, wobei “NNN” für eine dreistellige Zahl steht, die die Reihenfolge des Mappings angibt, d.h. die Mapping „001_Mapping1“ wird vor dem Mapping „002_Mapping2“ gemappt. „MAPPINGNAME“ ist der Name des Mappings.

REG_SZ:"expression"=""

Definiert eine Regular Expression, welche das Certificate Subject des Benutzers matchen soll. Es wird Perl Regular Expression Syntax verwendet.

REG_SZ:"formatter"=""

Definiert den Formatter, mit dem sich das Mapping-Ziel konfigurieren lässt.

Beispiele:

Smart Card Subject	**Expression**	**Formatter**	**LDAP User DN**	**Beschreibur**
cn=User1,o=comtarsia	(.*)	$1	cn=User1,o=comtarsia	1:1 Mapping
E=user1@comtarsia.com, cn=User1,o=comtarsia	E=([^,]*),(.*)	$2	cn=User1,o=comtarsia	Die Email-Adresse wir weggeschnit
cn=User1,o=comtarsia	([^,]*),([^,]*)	$1,ou=users,$2	cn=User1,ou=users,o=comtarsia	Einfügen ein OU in die DN
E=user1@comtarsia.com, cn=User1,o=comtarsia	E=([^,]*),[CcNn]=([^,]*),.*	$2	uid=$1,ou=personen,o=test	Ermitteln de Usernamens und umschreibe der DN

3.2.5. LDAP Systembenutzer

Die Benutzer-DN sowie das Passwort eines eventuell benötigten LDAP Systembenutzers (siehe Kapitel Static DN bis Smart Card Authentication) werden in der Registry angelegt.

Das Passwort wird verschlüsselt in der Registry abgelegt, wodurch das setzen dieses Wertes zwingend mittels des Comtarsia Management Console erfolgen muss.

REG_SZ:"systemUserDN"=""

Die Benutzer-DN des LDAP-Systembenutzers. Dieser Benutzer muss das Recht besitzen, nach dem jeweiligen Logon-Benutzer zu suchen.

Beispiel: „cn=SystemUser1,o=comtarsia“

REG_SZ:"systemUserPassword"=""

Das Passwort des LDAP-Systembenutzers. Zu Testzwecken kann in die Registry auch das Klartextpasswort geschrieben werden, in diesem Fall ist es mit „{CTP}“ zu prefixen.

3.3. LDAP Benutzerobjekt

Hier werden Parameter definiert, welche das LDAP-Benutzerobjekt betreffen.

REG_SZ:"userObjectClass"="Person"

REG_DWORD:"userObjectRequired"=00000001

Hiermit wird definiert, ob das LDAP-Benutzerobjekt vorhanden sein muss, um die Authentifizierung als erfolgreich zu werten.

Je nach ACL bzw. auch bei speziellen Direcory Servern kann es möglich sein, dass gar kein echtes Benutzerobjekt existiert bzw. der Benutzer nicht das Recht hat, dannach zu suchen. Dann kann dieser Wert auf „0“ gesetzt werden, wodurch dann bereits ein erfolgreicher LDAP-Bind als erfolgreiche Benutzerauthentifizierung gewertet wird.

REG_SZ:"userPasswordAttribute"="userPassword"

Definiert das LDAP-Attribut im Benutzerobjekt, welches das Benutzerpassword beinhält. Dies wird nur beim Passwort-Wechsel durch den SignOn Proxy verwendet, für die Authentifizierung erfolgt dieses Mapping am LDAP Server.

REG_DWORD:"useUTF8Password"=00000000

Hiermit kann konfiguriert werden, ob das Benutzerpasswort als UTF8-Encodierte Zeichenfolge (=1) order als 8bit ASCII (=0) zum LDAP-Server geschickt wird.

REG_DWORD:"dontSendOldPasswordOnChange"=00000000

Hier mit kann festgelegt werden, ob ein Passwordwechsel als LDAP-Replace-Operation (=0) oder als LDAP-Delete-Add-Operation(=1) durchgefuehrt wird. Dieser Konfigurationsparameter muss mit dem korrespondierenden Parameter am LDAP-Server abgestimmt werden.

REG_DWORD:"userQueryScope"=00000002

Definiert den LDAP Search Scope, mit welchem nach dem Benutzer gesucht wird.

Wert	Bedeutung
0	Base
1	One Level
2	Sub Tree

3.4. LDAP Gruppen

REG_DWORD:"groupTypes"=dword:00000007

Definiert eine Bitmaske über welche konfiguriert wird, welche Gruppentypen vom LDAP abgefragt werden sollen.

Wert	Gruppentyp
1	GroupOfNames
2	GroupOfUniqueNames
4	PosixGroup
8	Ibm-allGroups

REG_SZ:"groupQueryBase"=""

Per Default wird als Search Base für die Gruppensuche die BaseDN verwendet. Ist die „groupQueryBase“ konfiguriert, so wird dieser Wert anstelle der BaseDN als Search Base verwendet.

Beispiel: „ou=groups,o=comtarsia”

REG_DWORD:"groupQueryScope"=dword:00000002

Definiert das LDAP Query Scope fuer die Gruppensuche(n).

Wert	Bedeutung
0	Base
1	One Level
2	Sub Tree

REG_MULTI_SZ:"attributeBasedGroups"=[]

Hier kann eine Liste mit Attributen aus dem LDAP-Benutzerobjekt angegeben werden, welche vom LDAP ausgelesen werden und deren Inhalt dann als Gruppen dem Benutzer hinzugefügt wird.

Beispiel:

attributeBasedGroups="UserAGroups"

Im Benutzerobjekt gibt es ein Attribut mit dem Namen "UserAGroups", welches den Inhalt „Group1" hat.

REG_SZ:"groupFilter"=""

Hiermit kann ein Teil einer LDAP-Query angegeben werden, welcher dann bei jeder Gruppensuche mitangehängt wird.

3.5. Session Password

3.5.1. SetSessionPassword

Der Comtarsia SignOn Proxy kann das SessionPassword des Benutzers bei Bedarf in ein frei konfigurierbares LDAP-Attribut schreiben. Das Password kann optional verschlüsselt abgelegt werden.

Diese Funktion kann über einen Registry-Wert am Comtarsia SignOn Proxy 2008 aktiviert bzw. deaktiviert werden.

Als Verschlüsselungs-Algorithmus kommt der Advanced Encryption Standard mit 256 bit Schlüssellänge zur Anwendung.

Das Passwort wird vor der Verschlüsselung NULL-terminated und NULL-padded (auf 64 Zeichen). Die resultierenden verschlüesselten Daten werden dann Base64-Encoded im LDAP abgelegt. Dass Passwort zur Verschlüsselung wird in der Registry mit einem SignOn Proxy-internen Schlüssel AES-verschlüsselt abgelegt.

REG_DWORD:"setSessionPassword"=0

Hiermit kann diese Funktion aktiviert oder deaktiviert werden.

REG_SZ:"setSessionPasswordAttribute"=""

Definiert den Namen des LDAP Attributes, in welchen das Session Password geschrieben werden soll.

REG_DWORD:"setSessionPasswordEncryption"=0

Wert	Verschluesselung
0	Keine Verschlüsselung, das Passwort wird unverschlüsselt abgelegt.
1	Das Password wird mit AES256 verschlüsselt und anschliessend Base64 codiert abgelegt.

REG_SZ:"setSessionPasswordEncryptionKey"=""

Definiert den „Key“, der zur Verschlüsselung verwendet werden soll. Ist nur notwendig, wenn setSessionPasswordEncryption > 0. Der Key selbst wird verschlüsselt durch die Comtarsia Management Console in der Registry abgelgt.

REG_SZ:"setSessionPasswordCondition"="TRUE"

Definiert eine Bedingung, die erfüllt sein muss, wenn das Session Passwort geschrieben werden soll.

Gültige Werte für „Wahr“ sind “TRUE” oder eine Zahl ungleich 0.

Hier kann bei Bedarf auch eine Variable eingetragen werden, z.B: „%setSessionPassword%“, siehe auch Kapitel 5.

3.5.2. GetSessionPassword

Der SignOn Proxy kann das SessionPassword des Benutzers bei Bedarf von einem frei konfigurierbaren LDAP-Attribut lesen (Es werden nur Formate unterstützt, welche der SignOn Proxy auch bei SetSessionPassword schreiben kann, siehe Kapitel SetSessionPassword) und dieses Passwort dann für die Synchronisation auf die Agent-Systeme verwendet. Diese Funktion kann über einen Registry-Wert am Comtarsia SignOn Proxy 2008 aktiviert bzw. deaktiviert werden.

Das Passwort zur Entschlüsselung wird in der Registry mit einem SignOn Proxy-internen Schlüssel AES-verschlüsselt abgelegt.

Über ein frei konfigurierbares LDAP-Attribut des Benutzerobjektes kann gesteuert werden, ob das SessionPassword geschrieben (siehe Kapitel SetSessionPassword) oder gelesen werden soll.

Ueber einen Konfigurationswert lässt sich zusätzlich noch ein Filter auf den Logon Typ (Benutzer/Passwort oder SmartCard) erstellen.

REG_DWORD:"getSessionPassword"=0

Hiermit kann diese Funktion aktiviert oder deaktiviert werden.

REG_SZ:"getSessionPasswordAttribute"=""

Definiert den Namen des LDAP Attributes, aus welchem das Session Password gelsesen werden soll.

REG_DWORD:"getSessionPasswordEncryption"=0

Wert	Verschluesselung
0	Keine Verschlüsselung, das Passwort wird unverschlüsselt abgelegt.
1	Das Password wird mit AES256 verschlüsselt und anschliessend Base64 codiert abgelegt.

REG_SZ:"getSessionPasswordEncryptionKey"=""

Definiert den „Key“, der zur Entschlüsselung verwendet werden soll. Ist nur notwendig, wenn getSessionPasswordEncryption > 0. Der Key selbst wird verschlüsselt durch die Comtarsia Management Console in der Registry abgelgt.

REG_SZ:"getSessionPasswordCondition"="TRUE"

Definiert eine Bedingung, die erfüllt sein muss, wenn das Session Passwort geschrieben werden soll.

Gueltige Werte fuer „Wahr“ sind “TRUE” oder eine Zahl ungleich 0.

Hier kann bei Bedarf auch eine Variable eingetragen werden, z.B: „%setSessionPassword%“, siehe auch Kapitel 5.

3.5.3. SessionPassword Decrypt Example

Hier wird ein Beispiel aufgezeigt, wie man ein vom SignOn Proxy im LDAP gesetztes Session Passwort auf der Command Line (hier im Beispiel mit Cygwin unter Windows) decodieren kann. Dies soll als Grundlage dienen, um die benötigten Schritte bei Bedarf in Programmiersprachen oder Scripts umsetzen zu können.

Wenn der Encryption Key nicht genau 32 Zeichen hat, so wird er von der Management Console mit „0“en auf 32 Zeichen aufgefüllt, ist der Key länger als 32 Zeichen, so wird er auf 32 Zeichen gekürzt.

In key.txt wurde der Schlüssel „abcdefghijklmnopqrstuvwxyz123456“ abgelegt, welcher auch über die Comtarsia Management Console in das Feld SetSessionPasswordEncryptionKey eingetragen wurde.

1) $ *cat walletpassword.txt*

{aes}jSGmPwmwcgp6v4+LMAqsAD69ymFRpXArHuF0MwbLXOm1JoLRwNhB15znZ4MACQPVku0IZw34oyHNFYZsrkq4Nq3zEHRyN3ddGZrKd1Px0tRw0Qt5uc1BlyA8tHZyxQ7P7HuAqENpmMA3qdJgSo/8IVIAkAfyL8TB+oDFPjwmIS4Z/csAMHL1mQ3FHGP2jV

2) $ *dd if=walletpassword.txt of=walletpassword_wo_prefix.txt bs=1 skip=5*

216+0 records in

216+0 records out

216 bytes (216 B) copied, 0.005 s, 43 kB/s

3) $ *cat walletpassword_wo_prefix.txt*

jSGmPwmwcgp6v4+LMAqsAD69ymFRpXArHuF0MwbLXOm1JoLRwNhB15znZ4MACQPVku0IZw34oyHNFYZsrkq4Nq3zEHRyN3ddGZrKd1Px0tRw0Qt5uc1BlyA8tHZyxQ7P7HuAqENpmMA3qdJgSo/8IVIAkAfyL8TB+oDFPjwmIS4Z/csAMHL1mQ3FHGP2jV

4) Base64 decodieren

Hierfuer gibt's zwei Moeglichkeiten:

a) Base64 von den GNU coreutils

$ *base64 -d walletpassword_wo_prefix.txt > walletpassword.bin*

b) OpenSSL unterstützt nur eine begrenzte Zeilenlänge (OpenSSL verwendet als default 65 Zeichen pro Zeile, weniger geht auch), d.h. Man öffnet die Datei im Texteditor und Teilsie auf mindestens 4 Zeilen auf

$ *openssl base64 -d -in walletpassword_wo_prefix2.txt -out walletpassword.bin*

5) Aufteilen des Wertes auf Initialisierungsvektor (die ersten 16 Byte) und verschlüsselte Daten

$ *dd if=walletpassword.bin of=walletpassword.iv bs=1 count=16*

$ *dd if=walletpassword.bin of=walletpassword.crypt bs=1 skip=16*

6) $ *IV=`xxd -ps walletpassword.iv`*

2ddee6ce4f441e1dac789bb2688d161d

$ KEY=`xxd -ps -c 256 key.txt`

616263646566768696a6b6c6d6e6f707172737475767778797a313233343536

7) $ *openssl aes-256-cbc -nopad -d -K $KEY -iv $IV -in walletpassword.crypt -out walletpassword.plain*

8) $ *cat walletpassword.plain*

yyVXV863?%

4. Sync Domains

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\SignOnProxy\DomainSync\<DomainName>]

<DomainName> definiert den Namen der Synchronisations-Domain, wie er dann auch an den Client übertragen und von diesem angezeigt wird.

REG_DWORD:"type"=dword:0

Definiert den Sync Domain Type. Derzeit gibt es hier nur den Wert „0“.

REG_DWORD:"version"=1

Definiert die Agent-Version der Ziel-Domain. Ein SignOn Agent funktioniert auch im „SOG 2006“, allerdings sind damit auch Limitierungen verbunden, wie z.B. die maximale Anzahl an Synchronisations-Attributen (20 beim SOG 2006, unlimitiert bei SOG 2008).

Wert	Beschreibung
1	SignOn Gate 2006
2	SignOn Gate 2008

REG_DWORD:"hold"=0

Wird dieser Wert auf „1“ gesetzt, so ist die Sync Domain inaktiv und wird nicht verwendet.

REG_SZ:"syncPolicyAllow"="*"

Definiert eine List von Gruppen, wovon der Benutzer mindestens einer angehören muss, um in dieser Domain synchronisiert zu werden. Wildcard-Matching mit einem „*“ am Ende des jeweiligen Gruppennamens ist möglich.

REG_SZ:"syncPolicyDeny"=""

Definiert eine List von Gruppen, wovon der Benutzer keiner angehören darf, um in dieser Domain synchronisiert zu werden. Wildcard-Matching mit einem „*“ am Ende des jeweiligen Gruppennamens ist möglich.

[.\SOSProfile *\SignOnProxy\DomainSync\<DomainName>\<AgentName>]

<AgentName> definiert den Namen des jeweiligen Agents.

REG_DWORD:"port"=dword:2002

Der TCP-Port, auf dem der SignOn Proxy gebinded ist. 2002 Ist der Default Port.

REG_DWORD:"priority"=dword:0

Dieser Parameter ist derzeit nicht in Verwendung.

5. Variablen

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\Variables\<VariableEffectivePoint>\<VariableName>]

<VariableEffectivePoint> = "BeforeSync" oder "AfterSync"

Der „ VariableEffectivePoint“ bestimmt den Zeitpunkt, an welchem das jeweilige Variablen-Mapping durchgeführt werden soll.

Der Comtarsia SignOn Proxy bearbeitet jeden empfangenen Synchronisationsrequest in der folgenden Reihenfolge:

1) Empfang eines Datenpaketes vom Client (Logon Client, Web Client/Web Gateway)
2) LDAP-Prüfung des Benutzers

   VariableEffectivePoint: BeforeSync
3) Synchronisation der definierten Agents

   VariableEffectivePoint: AfterSync
4) Rücksenden eines Datenpaketes an den Client

<VariableName> muss in der Form „NNN_VARIABLENNAME“ sein, wobei “NNN” für eine dreistellige Zahl steht, die die Reihenfolge des Mappings angibt, d.h. die Variable „001_Var1“ wird vor der Variable „002_Var2“ gemappt. „VARIABLENNAME“ ist der Name der Variable.

REG_SZ:"displayName"=""

Definiert einen Anzeigenamen fuer die Variable, hat keine technische Funktion und kann als Art Kommentarfeld verwendet werden.

REG_SZ:"source"=""

Definiert die Quelle des Variablenmappings. Dies ist ein String, der ein oder mehrere Variablen (vordefinierte oder benutzerdefinierte) enthalten kann.

Beispiele:

- „%LDAPUser:sn%“

  Dies gibt an, dass das Attribut „sn“ des LDAP Benutzerobjektes eingesetzt werden soll. Durch das Verwenden des Attributes als Variable, wird dieses Attribut automatisch vom LDAP abgefragt.

  Beispiel: Hat das LDAP-Attribut „sn“ eines Benutzers den Wert „Meier“ so hat nun auch der Source den Wert „Meier“.

- „c:\homes\%USERNAME%“

  %USERNAME% ist eine interne SignOn Proxy Variable, welche automatisch immer zur Verfuegung steht und den Benutzername beinhaelt.

  Beispiel: Ist der Benutzername „User1“ so hat source den Wert „c:\homes\User1“.

- „%LDAPUser:__GROUP__[]%“

  Dies gibt an, dass die ueber LDAP ermittelten Benutzergruppen als source uebernommen werden. Das „[]“ gibt an, dass es sich um einen Multi-Value-Wert handelt.

REG_DWORD:"transmitDestination"=dword:0x0000

Definiert eine Bitmaske welche festlegt, wohin die Variable übertragen werden soll. Für rein lokale Variablen wird der wert „0“ gesetzt.

Wert	**Transmit Destination**
0x00	Keine, nut lokal
0x01	Logon Client
0x02	Web Gateway
0x04	SignOn Proxy
0x08	SignOn Agent

0x10	LDAP Directory Replicator

REG_MULTI_SZ:"transmitDestinationDomains"=[]

Definiert eine Liste von Agent-Domänen, an welche die jeweilige Variable übertragen werden soll. Damit diese Funktion aktiv ist, muss unter „transmitDestination“ das Bit „SignOn Agent“ gesetzt werden.

REG_DWORD:"mappingType"=0

Wert	Mapping Type
0	1:1 Mapping
1	Regular Expression Mapping

REG_SZ:"expression"=""

Eine Regular expression

REG_SZ:"formatter"=""

Der Formatter für das Ergebniss, z.B. „$1“

REG_DWORD:"index"=0

Wenn die Regular expression mehrfach matched, so kann hier angegeben weden, das wievielte Ergebniss genommen werden soll. Ist der Wert „0xffffffff“ gesetzt, so werden alle Ergebnise als Array uebernommen.

REG_DWORD:"flags"=0x2000000

Match Flags:

Flag	Name	Beschreibung
0x00000000	match_default	
0x00000001	match_not_bol	first is not start of line
0x00000002	match_not_eol	last is not end of line
0x00000004	match_not_bob	first is not start of buffer
0x00000008	match_not_eob	last is not end of buffer
0x00000010	match_not_bow	first is not start of word
0x00000020	match_not_eow	last is not end of word
0x00000040	match_not_dot_newline	\n is not matched by '.'

0x00000080	match_not_dot_null	'\0' is not matched by '.'
0x00000100	match_prev_avail	*--first is a valid expression
0x00000200	match_init	internal use
0x00000400	match_any	don't care what we match
0x00000800	match_not_null	string can't be null
0x00001000	match_continuous	each grep match must continue uninterupted from the previous one
0x00002000	match_partial	find partial matches
0x00004000	match_not_initial_null	don't match initial null
0x00008000	match_all	must find the whole of input even if match_any is set
0x00010000	match_perl	Use perl matching rules
0x00020000	match_posix	Use POSIX matching rules
0x00040000	match_nosubs	don't trap marked subs
0x00080000	match_extra	include full capture information for repeated captures
0x00100000	match_single_line	treat text as single line and ignor any \n's when matching ^ and $.
0x00200000	match_unused1	unused
0x00400000	match_unused2	unused
0x00800000	match_unused3	unused
0x00800000	match_max	

Format Flags:

Flag	**Name**	**Beschreibung**
0x00000000	format_perl	perl style replacement
0x01000000	format_sed	sed style replacement.
0x02000000	format_all	enable all extentions to sytax.
0x04000000	format_no_copy	don't copy non-matching segments.
0x08000000	format_first_only	Only replace first occurance.
0x10000000	format_is_if	internal use only.

0x20000000	format_literal	treat string as a literal

REG_DWORD:"multivalueAction"=0

Mit diesem Wert wird festgelegt, wie die Variable in das bereits bestehende Variablenpool eingefügt/entfernt werden soll.

Wert	Action
0	Override
1	Delete
2	DeleteValue
3	AddValue

REG_DWORD:"hold"=0

Ist dieser Wert aktiv, so wird das Variablenmapping nicht ausgeführt, und ist für den Proxy praktisch nicht existent. Diese Option dient hauptsaechlich zum Testen von Mappings, um einzelne Mappings schnell aktivieren oder deaktivieren zu können.

5.1. Interne Variablen

%USERNAME%

Enthält den Benutzernamen.

%__authType__%

Enthält den Authentifizierungstyp des Benutzer. Mögliche Werte:

Wert	Authentifizierungstyp
0	Benutzer/Passwort
1	Smart Card

%__DN__%

Diese interne Variable enthaelt die LDAP-DN des Benutzerobjektes.

%__GROUP__%

Diese interne Variable enthält alle Gruppen des Benutzers. Für Details zur Ermittlung der Gruppen aus dem LDAP siehe Kapitel

LDAP Gruppen. Diese Gruppenvariable kann über die Variablenfunktionen bearbeitet werden.

6. Profile Forwarding

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\SignOnProxy\ProfileForwarding\<ProfileForwardingName>]

REG_SZ:"profile"=""

REG_SZ:"source"=""

REG_SZ:"expression"=""

7. Logging

Der SignOn Proxy unterstuetzt derzeit zwei verschiedene Log-Ziele, welche unabhängig voneinander konfiguriert und aktiviert werden können. Grundsätzlich ist zu beachten, dass speziell beim Loglevel „Detail MSG“ eventuell noch in Kombination mit ein oder mehreren „Detail Log Flags“ zum Teil pro Logon Reuqest bis zu hundert Zeilen in das Log geschrieben werden.

7.1. Logging in eine Datei

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\Log]

REG_DWORD:"enable"=00000001

Mit diesem Parameter kann diese Loggingmethode aktiviert/deaktiviert werden.

REG_SZ:"logFileName"="%ProgramFiles%\\Comtarsia\\SignOn Solutions 2008\\log\\Comt%COMT_MODULE%.log"

Definiert den vollen Pfad der Log-Datei. Es können Environment-Variablen verwendet werden, wenn Sie im Kontext des jeweiligen Service-Benutzer zur Verfügung stehen, sowie die folgenden SignOn Proxy internen Variablen:

- %COMT_MODULE%

  Ist im Fall des SignOn Proxy immer “SOP”.

- %COMT_PROFILE_ID%

  Ist die Profile-ID des aktuellen in Verwendung befindlichen Profiles als dreistellige Zahl, z.B. „501“.

REG_DWORD:"logLevel"=00000004

Definiert das Log-Level. Als Default für den Produktiv-Betrieb wird „4“ empfohlen, bei Bedarf in Kombination mit Log Transactions. Alle niedrigeren Log-Level als das Konfigurierte sind immer automatisch enthalten, d.h. wird der Log-Level auf „2“ gesetzt, so werden alle „Error“ und alle „Exception“-Meldungen ausgegeben.

Wert	Log-Level

1	Error
2	Exception
3	Warning
4	Information
5	Detail Messages

REG_DWORD:"logMask"=00000000

Definiert eine Bitmaske, mit welcher sehr detaillierte Logausgaben für bestimmte Bereiche aktiviert werden können.

Wert	**Log-Bereich**
0x00000080	LDAP
0x00000100	LDAP SSL
0x00004000	Dump Config
0x00010000	Certificate Information
0x00020000	Dump Profile
0x00080000	Dump Variables
0x00200000	Variable Mapping

REG_DWORD:"logDetails"=0xFFFFFFFF

Definiert die Log-Details:

Wert	**Log-Details**
0x0	Keine Log Details
0x1	Date
0x2	Time
0x4	Prozess und Thread Ids
0x8	Source Postion
0xFFFFFFFF	Alle Details

REG_DWORD:"enableLogTransactions"=00000000

Aktiviert(=1) oder Deaktiviert(=0) Log-Transaktionen.

Sind die Log-Transaktionen aktiviert, so werden alle Log-Nachrichten, welche ein Log-Level haben, das hoeher als das konfigurierte ist, in einen internen Buffer geschrieben. Tritt dann spaeter bei der Bearbeitung noch ein „Error“ oder eine „Exception“ auf, so werden alle im Buffer befindlichen Nachrichten in die Datei geschrieben. Im Falle des SignOn Proxy umfasst eine Log-Transaktion genau eine Benutzer-Synchronisationsrequest.

REG_DWORD:"maxLogFileSize"=0x00a00000

Definiert die maximale Groesse der Log-Datei in Bytes. Überschreitet die Datei die definierte Grösse, so wird eine Log-Rotation durchgeführt.

REG_DWORD:"maxLogFileHistory"=00000001

Definiert wieviele Log-Histroien von Log-Rotationen aufgehoben werden sollen.

7.2. Logging in Syslog

[HKLM\SOFTWARE\Comtarsia\SOSProfile *\Log\SysLog]

Ist diese Logging-Methode aktiv, so werden alle Log-Nachrichten an einen Syslog-Server geschickt. (siehe RFC3164)

REG_DWORD:"enable"00000000

Mit diesem Parameter kann diese Loggingmethode aktiviert/deaktiviert werden.

REG_SZ:"host"=""

Definiert den SysLog Server. Hier kann ein Hostname oder eine IP-Adresse eingetragen werden.

REG_DWORD:"facility"=00000010

Definiert die SysLog facility.

Numerical Code	Facility
0	kernel messages
1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem

7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon
16	local use 0
17	local use 1
18	local use 2
19	local use 3
20	local use 4
21	local use 5
22	local use 6
23	local use 7

REG_DWORD:"logLevel"=00000000

Definiert den LogLevel. (siehe 6.1.)

Das Mapping des Comtarsia Log-Level auf die Syslog Severity erfolgt so:

Comtarsia Log-Level	**Severity Numerical Code**	**Severity**
1	3	Error: error conditions
2	3	Error: error conditions
3	4	Warning: warning conditions
4	5	Notice: normal but significant condition
5	6	Informational: informational messages

REG_DWORD:"logMask"=00000000

Definiert die Log-Details. (siehe Kapitel Logging in eine Datei)

REG_DWORD:"logDetails"=0x00000000

Definiert die Log-Details. (siehe Kapitel Logging in eine Datei)

7.3. Logfile-Format

```
A          B            C    D    E   F G                                          H    I  J
K

2010.10.08 11:15:59.195 1ACC:055C:00 <4:Comt::Prod::SOS2008::SOPConfig::readConfig:120:S0:C0>
Comtarsia SignOn Proxy 2008 Build 1.2.2.11

A          B            C    D    E
F  G                                  H   I  J  L                                                M
K

2010.10.08 11:09:28.698 09A4:0F34:00
<2:Comt::Prod::SOS2008::SOP::onServiceStart:62:S0:C0><Comt::Prod::SOS2008::SOP::onServiceStart:100>
The licence key is not valid!
```

A	Date
B	Time
C	Process ID in Hex
D	Thread ID in Hex
E	Log transaction
F	Message urgency
G	Source code module
H	Source code line
I	Message source
J	Message code
K	Message text
L	Source code module performing exception handling
M	Source code line performing exception handling

8. Installation / Komponenten

Die Installation der SignOn Proxy-Komponenten erfolgt durch den Installer nach „%PROGRAMFILES%\Comtarsia\SignOn Solutions 2008“.

A. Referenzen

RFC3164 - The BSD syslog Protocol http://tools.ietf.org/html/rfc3164