LAN Crypt Admin Windows

Multi-Policy-Support und Key-Tagging mit Version 11 der LAN Crypt Administration für Windows

Inhaltsverzeichnis

1 Funktionalität des Multi-Policy-Supports

Mit Version 11 der LAN Crypt Administration für Windows können Administratoren nun zusätzliche sekundäre Richtliniendateien für Nutzer einrichten. Dadurch können Nutzer, die LAN Crypt verwenden, Verschlüsselungsregeln von verschiedenen Administrationsstellen zugewiesen bekommen. Besonders praktisch ist dieses Feature beispielsweise für Arbeitnehmer, die in mehreren Arbeitsverhältnissen tätig sind und deren Arbeitgeber LAN Crypt für ihre Administration nutzen.

Bei der Nutzung dieses Features wird zwischen einer primären Richtlinie und mehreren sekundären Richtlinien unterschieden. Die primäre Richtlinie enthält die Informationen zur Ergänzung sekundärer Richtlinien, einschließlich der Speicherorte der sekundären Richtliniendateien und optional auch den Fingerabdruck des Zertifikats des Security Officers, von dem die jeweilige sekundäre Richtlinie stammt.

Wenn die primäre und die sekundären Richtliniendateien korrekt erstellt worden sind, werden diese beim Ladevorgang auf dem Nutzergerät geladen. Dabei werden alle Verschlüsselungsregeln aus den verschiedenen Richtliniendateien berücksichtigt.

1.1 Voraussetzungen für die Nutzung

1. Für die Nutzung wird mindestens Version 11 der LAN Crypt Administration für Windows benötigt. Bei der Installation muss das .NET API-Modul mitinstalliert worden sein.
2. Die Implementierung dieses Features erfordert eine Koordination zwischen dem Security Officer der primären Richtlinie und den Security Officers der sekundären Richtlinien. Es ist notwendig, Informationen über die Speicherorte der sekundären Richtlinien sowie optional den Fingerabdruck des Zertifikats des Security Officers, von dem die jeweilige sekundäre Richtlinie stammt, auszutauschen.
3. Da sekundäre Richtlinien im Gegensatz zu primären Richtlinien keinen automatischen Import von Benutzer- und Security Officer-Zertifikaten unterstützen, müssen die erforderlichen Zertifikate (Benutzer- und Security Officer-Zertifikat) für sekundäre Richtlinien manuell in den Windows Zertifikatsspeicher importiert oder auf einer Smartcard bereitgestellt werden.

1.2 Schritte zur Einrichtung sekundärer Richtlinien

Die Einrichtung von sekundären Richtlinien lässt sich in vier verschiedene Arbeitsschritte aufteilen:

1. Erstellung einer sekundären Richtliniendatei
2. Aktivierung des Multi-Policy-Supports in der Datenbank der LAN Crypt Administration
3. Verknüpfung von Nutzern mit sekundären Richtlinien
4. Konfigurieren der primären Richtliniendateien mit Verknüpfung zu sekundären Richtliniendateien

Alle Schritte lassen sich unkompliziert mithilfe der .NET-API ausführen. Dafür ist es jedoch notwendig, sich zunächst als Security Officer erfolgreich in die entsprechende Administrationsdatenbank einzuloggen.

Alle in diesem Dokument verwendeten Beispielskriptausschnitte können im Installations-Verzeichnis Ihres extrahierten Installationspakets in voller Länge eingesehen werden (LCAdmin\api\Examples\PowerShell\MultiPolicy).

PowerShell-Beispielskriptausschnitt für den Login in die Administrationsdatenbank:

```
# Sicherstellen, dass im 32-Bit-Modus gearbeitet wird
if ($env:Processor_Architecture -ne "x86")
{
  # Neustart im 32-Bit-Modus
  &"$env:windir\syswow64\windowspowershell\v1.0\powershell.exe" -noninteractive -noprofile -executionpolicy bypass -
file $myinvocation.Mycommand.path
  exit
}

# CLI- und .NET-DLLs laden
```

```
$dir = "C:\Program Files (x86)\Utimaco\u.trust LAN Crypt\Administration\"
$net = $dir + "LCAdminApiNet.dll"
Add-Type -Path $net

# High-Level API-Objekt initialisieren
$api = New-Object Conpal.LanCrypt.Admin.Api.LanCryptApi("SGLCSQLServer (ODBC Name)", "dbo (Database Owner)")

# An der Datenbank anmelden
$api.Database.Logon("Username", "Password", [Conpal.LanCrypt.Admin.Api.SqlDialect]::MSSQLServer) | Out-Null

# Master Security Officer (MSO) Objekt abrufen
$mso = $api.SecurityOfficers["Master Security Officer"]

# Als MSO anmelden
$mso.Logon() | Out-Null
```

PowerShell-Beispielskriptausschnitt zum Abmelden von der Administrationsdatenbank:

```
# Von Datenbank abmelden
$api.Database.Logoff
```

1.2.1 Erstellung sekundärer Richtliniendateien

Um die sekundären Richtliniendateien in die primären Richtlinien zu integrieren, muss der sekundäre Security Officer zunächst eine sekundäre Richtliniendatei erstellen. Dabei werden mithilfe der .NET API die Benutzer ausgewählt, deren Richtlinien als sekundäre Richtlinien exportiert werden sollen. Zusätzlich müssen ein Exportpfad für die Richtliniendatei angegeben und die Einstellungen für die Erstellung der sekundären Richtlinie festgelegt werden. Optional können der Datei Metadaten hinzugefügt werden, die in der LAN Crypt Administration des primären Security Officers angezeigt werden und ausschließlich im Abschnitt Client-Status einsehbar sind.

Die erstellte Richtliniendatei oder der Pfad einer zugänglichen Speicheradresse und das Zertifikat des sekundären Security Officers inklusive des Zertifikatfingerabdrucks können nun an den primären Security Officer übergeben werden.

PowerShell-Beispielskriptausschnitt zur Erstellung einer sekundären Richtliniendatei:

```
# Auswahl eines Nutzers, für den eine sekundäre Richtlinie erstellt werden soll
$secondaryCompanyLCUserName = "SecondaryCompanyUserName"
$secondaryCompanyLCUser = $api.Users[$secondaryCompanyLCUserName]

# Das Ausgabeverzeichnis für die sekundäre Richtlinie definieren
$outputDirectory = "\\server\path\to\policy\output\directory\at\secondary\company"

# Einstellungen zum Erstellen der Profile der sekundären Richtlinie festlegen
$settings = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectory, $secondaryCompanyLCUser)

# Die Richtlinie als sekundäre Richtlinie markieren
$settings.MultiPolicySecondaryProfile = $true

# Optionale Meta-Informationen für die sekundäre Richtlinie
# Diese werden in der LAN Crypt Administration zusammen mit den Meta-Informationen aller geladenen
# Richtlinien angezeigt und sind ausschließlich im Abschnitt Client-Status einsehbar.
```

```
$settings.MetaInformation = "Meta information for secondary policy - $(Get-Date -format 'u')"

# Die sekundäre Richtlinie basierend auf den festgelegten Einstellungen schreiben
$api.Profiles.WriteProfiles($settings)
```

1.2.2 Aktivierung des Multi-Policy-Supports in der LAN Crypt Datenbank

Die Integration sekundärer Richtlinien erfordert zunächst das Eintragen der Unternehmen, die den primären Security Officer mit den entsprechenden Richtliniendateien versorgen.

Hierfür muss die Konfiguration der Administrationsdatenbank mittels der .NET API angepasst werden, um die Unterstützung für sekundäre Richtlinien zu aktivieren. In diesem Schritt werden sowohl die Unternehmen als auch die zugehörigen Security Officer in die Datenbank aufgenommen. Zur Erhöhung der Sicherheit kann zudem der Fingerabdruck des jeweiligen Security Officer-Zertifikats hinterlegt werden, der beim Ladevorgang von Richtliniendateien überprüft wird. Diese Maßnahme wird ausdrücklich empfohlen.

PowerShell-Beispielskriptausschnitt zur Aktivierung der MultiPolicy-Funktionalität und zum Hinzufügen von Unternehmenseinträgen in der Administrationsdatenbank:

```
# Multi-Policy-Funktionalität in LAN Crypt Administration aktivieren, um primäre Richtlinien zu erstellen, die für
# Multi-Policy geeignet sind.
# Diese Operation schreibt einen Eintrag in die AppConfig-Tabelle in der LAN Crypt Datenbank.
$api.MultiPolicy.EnableMultiPolicyPrimary()

# Einen Namen für das Unternehmen definieren, das sekundäre Profile liefert
$secondaryCompanyName = "SecondaryCompanyName"

# Einen Eintrag für das Unternehmen in der LAN Crypt Datenbank erstellen
$organization = $api.MultiPolicy.Organizations.Create($secondaryCompanyName)

$secondarySoName = "SecondaryCompanySoName"

# Einen SO-Eintrag für das Unternehmen in der LAN Crypt Datenbank erstellen
$secondarySo = $organization.SecurityOfficers.Create($secondarySoName)

# Die Details des SO-Zertifikats definieren, das verwendet wird, um die sekundäre Richtlinie zu signieren
# Wenn die sekundäre Richtlinie mit einem Zertifikat signiert wird, das einen anderen Fingerabdruck besitzt,
# wird sie nicht geladen.
# Wenn dieser Parameter weggelassen wird, wird die sekundäre Richtlinie unabhängig von dem SO-Zertifikat,
# mit dem sie signiert wurde, geladen.
$secondarySoCertName = "SecondaryCompanySoCertName"
$fingerprint = "SecondaryCompanySoCertFingerprint"
$validFrom = "2024-01-01"
$validTo = "2074-01-01"

# Den Eintrag des Unternehmens in die LAN Crypt Datenbank schreiben
$secondarySo.Certificates.Create($secondarySoCertName, $fingerprint, $validFrom, $validTo) | Out-Null
```

1.2.3 Verknüpfung von Nutzern mit sekundären Richtlinien

Da nun die sekundären Richtlinien einschließlich der Zertifikate der sekundären Security Officers vorliegen und die Unternehmen in die für die Multi-Policy-Funktionalität konfigurierte

Datenbankstruktur eingetragen sind, können diese Informationen gebündelt in Profilen zusammengefasst und mit den Nutzern von primären Richtlinien verknüpft werden.

PowerShell-Beispielskriptausschnitt zur Verknüpfung eines Nutzers mit einem Unternehmenseintrag und sekundärer Richtlinie:

```
# Den primären Benutzer auswählen, für den eine primäre Richtlinie mit Einträgen für sekundäre Richtlinien
# erstellt werden soll
$primaryLCUserName = "PrimaryCompanyLCUserName"
$primaryLCUser = $api.Users[$primaryUserName]

# Einen Namen für einen sekundären Benutzer angeben
# Dieser sekundäre Benutzer wird mit dem primären Nutzer verknüpft und enthält die Verbindungen zu
# sekundären Richtlinien
$secondaryLCUserName = $primaryUserName + "_sec"

# Das im letzten Schritt erstellte Unternehmen abrufen
$secondaryOrganizationName = "SecondaryCompanyName"
$organization = $api.MultiPolicy.Organizations[$secondaryOrganizationName]

# Einen Eintrag zur Verknüpfung des Nutzers mit dem Unternehmen in der LAN Crypt Datenbank erstellen
$secondaryLCUser = $organization.Users.Create($secondaryUserName, $primaryUser)

# Den Pfad definieren, wo die sekundäre Richtlinie verfügbar ist
$pathToSecondaryPolicy = "\\path\to\secondary\policy\secondaryPolicyFileName.xml.bz2"

# Einen Eintrag für die sekundäre Richtlinie für den sekundären Benutzer in der LAN Crypt Datenbank erstellen
# Der sekundäre Nutzer sammelt sekundäre Richtlinien und ist mit dem primären Nutzer so verknüpft, dass alle #
Richtlinien auf diesen übertragen werden.
# Hinweis: Ein Benutzer kann mehrere Einträge für sekundäre Richtlinien haben
$secondaryLCUser.Profiles.Create($pathToSecondaryPolicy) | Out-Null
```

1.2.4 Konfiguration primärer Richtliniendateien mit sekundären Richtlinien

Nachdem alle vorangegangenen Schritte abgeschlossen sind, können nun primäre Richtliniendateien für Benutzer konfiguriert werden. Diese Richtlinien laden automatisch die mit ihnen verknüpften sekundären Richtlinien samt deren Metainformationen und Unternehmenszugehörigkeit.

Auch wenn primäre Richtlinien über das GUI erstellt werden, werden die verknüpften sekundären Richtlinien für die Benutzer automatisch mitgeladen. Allerdings können Metainformationen für die primären Richtlinien – im Gegensatz zur Nutzung der .NET API – nicht vergeben werden.

Wenn im Aktivierungsschritt die Option EnableMultiPolicyPrimary() nicht gesetzt wird, werden ausschließlich reguläre (primäre) Richtlinien erstellt, ohne sekundäre Richtlinien zu verknüpfen. Dies gilt sowohl für die Erstellung von Richtlinien über die .NET API als auch über das GUI.

PowerShell-Beispielskriptausschnitt zur Konfiguration einer primären Richtliniendateien:

```
# Überprüft, ob Multi-Policy-Support in der Datenbank aktiviert ist.
# Falls nicht, wird das Skript an dieser Stelle beendet.
if(-not $api.MultiPolicy.IsMultiPolicyPrimaryEnabled())
{
	$api.Database.Logoff

	exit 0
}

# Primäre Richtlinien wie gewohnt schreiben. Die Einträge für sekundäre Richtlinien werden automatisch
# hinzugefügt, sofern sie in der LAN Crypt konfiguriert sind.
# Einen primären LAN Crypt Benutzer auswählen, um eine primäre Richtliniendatei zu schreiben
$primaryLCUserName = "PrimaryCompanyUserName"
$primaryLCUser = $api.Users[$primaryLCUserName]

$outputDirectory = "\\server\path\to\policy\output\directory\at\primary\company"

# ProfilesCreateData mit dem primären Benutzer erstellen
$settingsPrimaryPolicy = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectory, $primaryLCUser)

# Optionale Meta-Informationen für die primäre Richtlinie
# Diese werden in der Administration zusammen mit den Meta-Informationen aller geladenen Richtlinien
# angezeigt und sind ausschließlich im Abschnitt Client-Status einsehbar.
$settingsPrimaryPolicy.MetaInformation = "Meta information for primary policy - $(Get-Date -format 'u')"

$api.Profiles.WriteProfiles($settingsPrimaryPolicy)
```

Wenn Richtlinien für eine Gruppe erstellt werden und Nutzer, die Mitglied dieser Gruppe sind, eigene verknüpfte sekundäre Richtlinien haben, werden diese sekundären Richtlinien für diese individuellen Nutzer automatisch mitgeladen, jedoch nicht für die gesamte Gruppe.

Sekundäre Richtlinien werden nicht geladen, wenn sie Regeln bestehender Richtlinien überschreiben würden.

2 Key-Tagging Funktionalität

Version 11 der LAN Crypt Administration für Windows bietet zusätzlich eine verbesserte Schlüsselverwaltung, die durch die Nutzung von Tags erweitert wurde. Mit diesen Tags können bestimmte Schlüssel markiert werden, was das Ein- und Ausschließen von Schlüsseln beim Zuweisen von Richtlinien zu Benutzern erheblich erleichtert.

Diese Funktion ist besonders nützlich beim Einsatz des Multi-Policy-Supports und der Erstellung sekundärer Richtlinien. Dabei können gezielt Schlüssel, die beispielsweise durch

eine Gruppenzugehörigkeit mit einem Profil verknüpft sind, ausgeschlossen werden. So behält ein Unternehmen, das sekundäre Richtlinien bereitstellt, die volle Kontrolle über die gemeinsam genutzten Schlüssel und die damit verbundenen Verschlüsselungsregeln.

Key-Tagging ist ausschließlich über die .NET-API verfügbar. Das Ein- und Ausschließen von Schlüsseln und den dazugehörigen Verschlüsselungsregeln ist nur beim Erstellen von Richtliniendateien über die API möglich.

2.1 Voraussetzungen für die Nutzung

Für die Nutzung wird mindestens Version 11 der LAN Crypt Administration für Windows benötigt. Bei der Installation muss das .NET API-Modul mitinstalliert worden sein.

2.2 Verwendung von Key-Tagging

Die Verwendung von Key-Tags erfolgt in zwei Schritten: Zunächst wird ein Tag einem Schlüssel zugewiesen. Anschließend können die Schlüssel und ihre zugehörigen Verschlüsselungsregeln beim Erstellen von (sekundären) Richtliniendateien anhand der zugewiesenen Tags gefiltert werden.

Alle Schritte lassen sich unkompliziert mithilfe der .NET-API ausführen. Dafür ist es jedoch notwendig, sich zunächst als Security Officer erfolgreich in die entsprechende Administrationsdatenbank einzuloggen.

Alle in diesem Dokument verwendeten Beispielskriptausschnitte können im Installations-Verzeichnis Ihres extrahierten Installationspakets in voller Länge eingesehen werden (LCAdmin\api\Examples\PowerShell).

PowerShell-Beispielskriptausschnitt für den Login in die Administrationsdatenbank:

```
# Sicherstellen, dass im 32-Bit-Modus gearbeitet wird
if ($env:Processor_Architecture -ne "x86")
{
  # Neustart im 32-Bit-Modus
  &"$env:windir\syswow64\windowspowershell\v1.0\powershell.exe" -noninteractive -noprofile -executionpolicy bypass -file $myinvocation.Mycommand.path
  exit
}

# CLI- und .NET-DLLs laden
$dir = "C:\Program Files (x86)\Utimaco\u.trust LAN Crypt\Administration\"
```

```
$net = $dir + "LCAdminApiNet.dll"
Add-Type -Path $net

# High-Level API-Objekt initialisieren
$api = New-Object Conpal.LanCrypt.Admin.Api.LanCryptApi("SGLCSQLServer (ODBC Name)", "dbo (Database Owner)")

# An der Datenbank anmelden
$api.Database.Logon("Username", "Password", [Conpal.LanCrypt.Admin.Api.SqlDialect]::MSSQLServer) | Out-Null

# Master Security Officer (MSO) Objekt abrufen
$mso = $api.SecurityOfficers["Master Security Officer"]

# Als MSO anmelden
$mso.Logon() | Out-Null
```

PowerShell-Beispielskriptausschnitt zum Abmelden von der Administrationsdatenbank:

```
# Von Datenbank abmelden
$api.Database.Logoff
```

2.2.1 Erstellung und Verknüpfung von Key-Tags

Der folgende PowerShell-Skriptabschnitt zeigt, wie ein Tag mit dem Namen NotForExternalUsage erstellt und anschließend dem Schlüssel NEW_KEY zugewiesen wird. Durch diese Verknüpfung kann der Schlüssel sowie die zugehörige Verschlüsselungsregeln bei der Erstellung der Richtliniendatei im nächsten Schritt gezielt ein- oder ausgeschlossen werden.

PowerShell-Beispielskriptausschnitt zum Erstellen und Verknüpfen von Key-Tags:

```
# Erstellt einen Tag für einen Schlüssel
$KeyTag = $api.tag.create("NotForExternalUsage")

# Name des Schlüssels, der mit dem Tag verknüpft werden soll
$KeyName = "NEW_KEY";

# Ruft das Schlüsselobjekt anhand des angegebenen Schlüsselnamens ab
$key = $api.Keys.Item($KeyName)

# Fügt den erstellten Tag dem Schlüssel hinzu
$key.AddTag($KeyTag)
```

2.2.2 Ein- und Ausschließen von Key-Tags

Beim Erstellen von sekundären Richtlinien besteht nun die Möglichkeit, Schlüssel und deren zugehörige Verschlüsselungsregeln auszuschließen, indem die mit ihnen verknüpften Tags gefiltert werden.

Tags können auf zwei verschiedene Arten gefiltert werden. Im Beispiel wird gezeigt, wie einzelne Schlüssel/ Verschlüsselungsregeln durch die Methode TagsToExclude.Add($tagToExclude) ausgeschlossen werden können. Alternativ besteht auch die Möglichkeit, bestimmte Schlüssel

oder Richtlinien explizit mit TagsToInclude.Add($tagToInclude) einzuschließen und gleichzeitig alle anderen auszuschließen.

PowerShell-Beispielskriptausschnitt zur Erstellung einer sekundären Richtlinie unter Ausschluss von getaggten Schlüsseln/ Verschlüsselungsregeln:

```
# Finde den LAN Crypt Benutzer, für den eine sekundäre Richtlinie erstellt werden soll
$secondaryCompanyLCUserName = "UserName"
$secondaryCompanyLCUser = $api.Users[$secondaryCompanyLCUserName]

# Verzeichnis für die Ausgabe
$outputDirectoryWithTagFiltering = "C:\temp\SecondaryWithTagFiltering\"
$settingsWithTagFiltering = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectoryWithTagFiltering,
$secondaryCompanyLCUser);

# Markiere die Richtlinie als sekundäre Richtlinie
$settingsWithTagFiltering.MultiPolicySecondaryProfile = $true

# Optionale und beliebige Metainformationen für die sekundäre Richtlinie
# werden in der Client-Status-App zusammen mit den Metainformationen aller geladenen Richtlinien angezeigt
$settingsWithTagFiltering.MetaInformation = "Meta information for secondary policy - $(Get-Date -format 'u')"

# Tags zum Ausschließen
$tagToExclude = $api.tag["NotForExternalUsage"]
$settingsWithTagFiltering.TagsToExclude.Add($tagToExclude)

$api.Profiles.WriteProfiles($settingsWithTagFiltering)
```

Imprint

Copyright 2024	Utimaco IS GmbH Germanusstr. 4 D-52080 Aachen Germany
Phone	+49 (0)241 / 1696-200
Fax	+49 (0)24w1 / 1696-199
Internet	utimaco.com
e-mail	hsm@utimaco.com
Document Version	1.0.0
Date	<2024-06-11>