LAN Crypt Admin Windows

Multi-policy support and Key-Tagging with version 11 of LAN Crypt Administration for Windows

Table of contents

1 Functionality of the multi-policy support

With version 11 of LAN Crypt Administration for Windows, administrators can now set up additional secondary policy files for users. This allows users who use LAN Crypt to be assigned encryption rules from different administration centers. This feature is particularly useful, for example, for employees who work in several employment relationships and whose employers use LAN Crypt for their administration.

When using this feature, a distinction is made between a primary policy and several secondary policies. The primary policy contains the information to supplementary secondary policies, including the locations of the secondary policy files and optionally also the fingerprint of the certificate of the security officer from whom the respective secondary policy originates.

If the primary and secondary policy files have been created correctly, they are loaded to the user device during the loading process. All encryption rules from the various policy files are considered.

1.1 Prerequisites for use

1. At least version 11 of LAN Crypt Administration for Windows is required for use. The .NET API module must have been installed during installation.

2. The implementation of this feature requires coordination between the Security Officer of the primary policy and the Security Officers of the secondary policies. It is necessary to exchange information about the locations of the secondary policies and optionally the fingerprint of the certificate of the security officer from which the respective secondary policy originates.
3. Since secondary policies, unlike primary policies, do not support automatic import of user and security officer certificates, the required certificates (user and security officer certificate) for secondary policies must be manually imported into the Windows certificate store or provided on a smartcard.

1.2 Setting up secondary policies

Setting up secondary policies can be divided into four different steps:

1. Creation of a secondary policy file
2. Activation of multi-policy support in the LAN Crypt Administration database
3. Linking users with secondary policies
4. Configuration of the primary policy files with links to secondary policy files

All steps can be carried out easily using the .NET API. However, it is first necessary to successfully log into the corresponding administration database as a security officer.

All example script excerpts used in this document can be viewed in full length in the installation directory of your extracted installation package (LCAdmin\api\Examples\PowerShell).

PowerShell sample script snippet for logging into the administration database:

```
# Ensure that it is running in 32-bit mode
if ($env:Processor_Architecture -ne "x86") {
   # Restart in 32-bit mode
   &"$env:windir\syswow64\windowspowershell\v1.0\powershell.exe" -noninteractive -noprofile -executionpolicy bypass -
file $myinvocation.Mycommand.path
   exit
}

# Load CLI and .NET DLLs
$dir = "C:\Program Files (x86)\Utimaco\u.trust LAN Crypt\Administration\"
$net = $dir + "LCAdminApiNet.dll"
Add-Type -Path $net

# Initialize high-level API object
$api = New-Object Conpal.LanCrypt.Admin.Api.LanCryptApi("SGLCSQLServer (ODBC Name)", "dbo (Database Owner)")
```

```
# Log on to the database
$api.Database.Logon("Username", "Password", [Conpal.LanCrypt.Admin.Api.SqlDialect]::MSSQLServer) | Out-Null

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

# Log on as MSO
$mso.Logon() | Out-Null
```

PowerShell sample script snippet for logging out of the administration database:

```
# Log out of database
$api.Database.Logoff
```

1.2.1 Creation of secondary policy files

To integrate the secondary policy files into the primary policies, the secondary security officer must first create a secondary policy file. The .NET API is used to select the users whose policies are to be exported as secondary policies. In addition, an export path for the policy file must be specified and the settings for creating the secondary policy must be defined. Optionally, metadata can be added to the file, which is displayed in the LAN Crypt Administration of the primary security officer and can only be viewed in the Client status section.

The created policy file or the path of an accessible storage address and the certificate of the secondary security officer including the certificate fingerprint can now be transferred to the primary security officer.

PowerShell sample script snippet for creating a secondary policy file:

```
# Selection of a user for whom a secondary policy is to be created
$secondaryCompanyLCUserName = "SecondaryCompanyUserName"
$secondaryCompanyLCUser = $api.Users[$secondaryCompanyLCUserName]

# Defining the output directory for the secondary directive
$outputDirectory = "\\server\path\to\policy\output\directory\at\secondary\company"

# Define settings for creating the profiles of the secondary policy
$settings = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectory, $secondaryCompanyLCUser)

# Mark the directive as a secondary directive
$settings.MultiPolicySecondaryProfile = $true

# Optional meta information for the secondary policy
# These are displayed in LAN Crypt Administration together with the meta information of all loaded
# Policies and can only be viewed in the Client Status section.
$settings.MetaInformation = "Meta information for secondary policy - $(Get-Date -format 'u')"

# Writing the secondary policy based on the defined settings
$api.Profiles.WriteProfiles($settings)
```

1.2.2 Activating multi-policy support in the LAN Crypt database

The integration of secondary policies first requires the entry of the companies that supply the primary Security Officer with the corresponding policy files.

To do this, the configuration of the administration database must be adapted using the .NET API to activate support for secondary policies. In this step, both the companies and the associated security officers are added to the database. To increase security, the fingerprint of the respective security officer certificate can also be stored, which is checked when loading policy files. This measure is strongly recommended.

PowerShell sample script snippet for activating the MultiPolicy functionality and adding company entries in the administration database:

```
# Activate multi-policy functionality in LAN Crypt Administration to create primary policies that are suitable for
# suitable for multi-policy.
# This operation writes an entry to the AppConfig table in the LAN Crypt database.
$api.MultiPolicy.EnableMultiPolicyPrimary()

# Define a name for the company that provides secondary profiles
$secondaryCompanyName = "SecondaryCompanyName"

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

$secondarySoName = "SecondaryCompanySoName"

# Create an entry for the company in the LAN Crypt database
$secondarySo = $organization.SecurityOfficers.Create($secondarySoName)

# Define the details of the SO certificate used to sign the secondary policy
# If the secondary policy is signed with a certificate that has a different fingerprint, it will not be loaded.
# If this parameter is omitted, the secondary policy becomes independent of the SO certificate,
# with which it was signed is loaded.
$secondarySoCertName = "SecondaryCompanySoCertName"
$fingerprint = "SecondaryCompanySoCertFingerprint"
$validFrom = "2024-01-01"
$validTo = "2074-01-01"

# Write the company's entry in the LAN Crypt database $secondarySo.Certificates.Create($secondarySoCertName,
$fingerprint, $validFrom, $validTo) | Out-Null
```

1.2.3 Linking users with secondary policies

Now that the secondary policies, including the certificates of the secondary security officers, are available and the companies are entered in the database structure configured for the multi-policy functionality, this information can be bundled into profiles and linked to the users of primary policies.

PowerShell example script snippet for linking a user with a company entry and secondary policy:

```
# Select the primary user for whom a primary policy with entries for secondary policies is to be created.
$primaryLCUserName = "PrimaryCompanyLCUserName"
$primaryLCUser = $api.Users[$primaryUserName]

# Specify a name for a secondary user
# This secondary user will be linked to the primary user and will contain the connections to secondary policies
$secondaryLCUserName = $primaryUserName + "_sec"

# Retrieve the company created in the last step
$secondaryOrganizationName = "SecondaryCompanyName"
$organization = $api.MultiPolicy.Organizations[$secondaryOrganizationName]

# Create an entry to link the user to the company in the LAN Crypt database
$secondaryLCUser = $organization.Users.Create($secondaryUserName, $primaryUser)

# Define the path where the secondary policy is available
$pathToSecondaryPolicy = "\\path\to\secondary\policy\secondaryPolicyFileName.xml.bz2"

# Create an entry for the secondary policy for the secondary user in the LAN Crypt database
# The secondary user collects secondary policies and is linked to the primary user so that all
# policies are transferred to it.
# Note: A user can have several entries for secondary policies$secondaryLCUser.Profiles.Create($pathToSecondaryPolicy) |
Out-Null
```

1.2.4 Configuration of primary policy files with secondary policies

Once all the previous steps have been completed, primary policy files can now be configured for users. These policies automatically load the secondary policies linked to them, including their meta information and company affiliation.

Even if primary policies are created via the GUI, the linked secondary policies for the users are also loaded automatically. However, unlike when using the .NET API, meta information cannot be assigned for the primary policies.

If the EnableMultiPolicyPrimary() option is not set in the activation step, only regular (primary) policies are created without linking secondary policies. This applies both to the creation of policies via the .NET API and via the GUI.

PowerShell sample script snippet for configuring a primary policy file:

```
# Checks whether multi-policy support is activated in the database.
if(-not $api.MultiPolicy.IsMultiPolicyPrimaryEnabled())
{
        $api.Database.Logoff

        exit 0
}
```

```
# Write primary policies as usual. The entries for secondary policies are added automatically
# added if they are configured in the LAN Crypt.
# Select a primary LAN Crypt user to write a primary policy file
$primaryLCUserName = "PrimaryCompanyUserName"
$primaryLCUser = $api.Users[$primaryLCUserName]

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

# Create ProfilesCreateData with the primary user
$settingsPrimaryPolicy = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectory, $primaryLCUser)

# Optional meta information for the primary policy
# These are displayed in the administration together with the meta information of all loaded policies
# and can only be viewed in the Client status section.
$settingsPrimaryPolicy.MetaInformation = "Meta information for primary policy - $(Get-Date -format 'u')"

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

If policies are created for a group and users who are members of this group have their own linked secondary policies, these secondary policies are automatically loaded for these individual users, but not for the entire group.

Secondary policies are not loaded if they would overwrite the rules of existing policies would be overwritten.

2 Key-Tagging functionality

Version 11 of LAN Crypt Administration for Windows introduces enhanced key management, now extended through the use of tags. These tags allow specific keys to be marked, significantly simplifying the inclusion or exclusion of keys when assigning policies to users.

This feature is particularly useful when utilizing multi-policy support and creating secondary policies. It enables the targeted exclusion of keys that are linked to a profile. As a result, a company deploying secondary policies retains full control over shared keys and the associated encryption rules.

Key tagging is only available via the .NET API. The inclusion and exclusion of keys and their associated policies can only be managed when creating policy files through the API.

2.1 Prerequisites for use

At least version 11 of LAN Crypt Administration for Windows is required for use. The .NET API module must have been installed during installation.

2.2 Use of Key-Tagging

The use of key tags involves two steps: First, a tag is assigned to a key. Then, when creating (secondary) policy files, the keys and their associated encryption rules can be filtered based on the assigned tags.

All steps can be carried out easily using the .NET API. However, it is first necessary to successfully log into the corresponding administration database as a security officer.

All example script excerpts used in this document can be viewed in full length in the installation directory of your extracted installation package (LCAdmin\api\Examples\PowerShell).

PowerShell sample script snippet for logging into the administration database:

```
# Ensure that it is running in 32-bit mode
if ($env:Processor_Architecture -ne "x86") {
   # Restart in 32-bit mode
   &"$env:windir\syswow64\windowspowershell\v1.0\powershell.exe" -noninteractive -noprofile -executionpolicy bypass -
file $myinvocation.Mycommand.path
   exit
}

# Load CLI and .NET DLLs
$dir = "C:\Program Files (x86)\Utimaco\u.trust LAN Crypt\Administration\"
$net = $dir + "LCAdminApiNet.dll"
Add-Type -Path $net

# Initialize high-level API object
$api = New-Object Conpal.LanCrypt.Admin.Api.LanCryptApi("SGLCSQLServer (ODBC Name)", "dbo (Database Owner)")

# Log on to the database
$api.Database.Logon("Username", "Password", [Conpal.LanCrypt.Admin.Api.SqlDialect]::MSSQLServer) | Out-Null

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

# Log on as MSO
$mso.Logon() | Out-Null
```

PowerShell sample script snippet for logging out of the administration database:

```
# Log out of database
$api.Database.Logoff
```

2.2.1 Creation and linking of Key-Tags

The following PowerShell script snippet demonstrates how to create a tag named NotForExternalUsage and then assign it to the key NEW_KEY. This link allows the key and its associated encryption rule to be specifically included or excluded when creating the policy file in the next step.

PowerShell sample script snippet for creating and linking Key-Tags:

```
# Creates a tag for a key
$KeyTag = $api.tag.create("NotForExternalUsage")

# Name of the key to be linked with the tag
$KeyName = "NEW_KEY";

# Retrieves the key object based on the specified key name
$key = $api.Keys.Item($KeyName)

# Adds the created tag to the key
$key.AddTag($KeyTag)
```

2.2.2 Including and excluding Key-Tags

When creating secondary policies, it is now possible to exclude keys and their associated encryption rules by filtering the tags linked to them.

Tags can be filtered in two different ways. The example shows how individual keys/encryption rules can be excluded using the method TagsToExclude.Add($tagToExclude). Alternatively, there is also the option to explicitly include certain keys with TagsToInclude.Add($tagToInclude) while excluding all others.

PowerShell sample script snippet for creating a secondary policy with exclusion of tagged keys/encryption rules:

```
# Find the LAN Crypt user for whom a secondary policy is to be created
$secondaryCompanyLCUserName = "UserName"
$secondaryCompanyLCUser = $api.Users[$secondaryCompanyLCUserName]

# Directory for output
$outputDirectoryWithTagFiltering = "C:\temp\SecondaryWithTagFiltering\"
$settingsWithTagFiltering = New-Object Conpal.LanCrypt.Admin.Api.ProfilesCreateData($outputDirectoryWithTagFiltering,
$secondaryCompanyLCUser);

# Mark the policy as a secondary policy
$settingsWithTagFiltering.MultiPolicySecondaryProfile = $true
```

```
# Optional and arbitrary metadata for the secondary policy
# will be displayed in the client status app along with metadata of all loaded policies
$settingsWithTagFiltering.MetaInformation = "Meta information for secondary policy - $(Get-Date -format 'u')"

# Tags to exclude
$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>