One Identity Manager 8.0.2

Password Capture Agent Administration Guide

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

One Identity Manager Password Capture Agent Administration Guide
Updated - September 2018
Version - 8.0.2

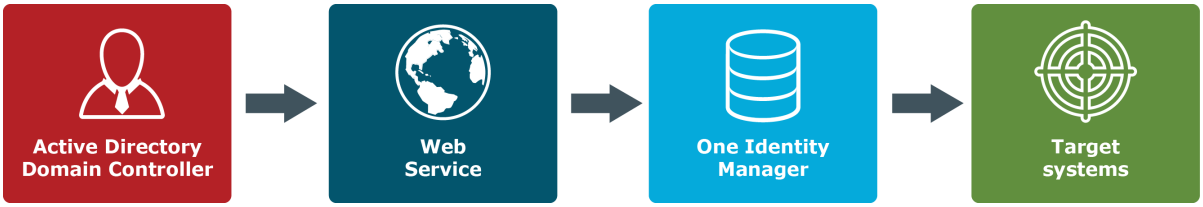
Contents

The One Identity Manager Password Capture Agent

The Password Capture Agent allows you to synchronize user passwords between Active Directory domains managed by One Identity Manager and other connected target systems. The Password Capture Agent tracks changes to user passwords in the source Active Directory domain and provides that information to the Webservice (an optional component of One Identity Manager), which in turn synchronizes the changes with target connected data systems by using the password templates you specified. To synchronize passwords, you must install the Password Capture Agent on each domain controller in the Active Directory domain you want to use, as a source for the password synchronization operations.

The following diagram shows how the password synchronization feature of One Identity Manager works.

Figure 1: How the Password Synchronization feature works



Automated Password Synchronization

If your enterprise environment has multiple data management systems, each having its own password policy and dedicated user authentication mechanism, you may face one or more of the following issues:

- Because users have to remember multiple passwords, they may have difficulty managing them. Some users may even write down their passwords. As a result, passwords can be easily compromised.

- Each time users forget one or several of their numerous access passwords, they have to ask administrators for password resets. This increases operational costs and translates into a loss of productivity.
- There is no way to implement a single password policy for all of the data management systems. This tool impacts productivity, as users have to log on to each data management system separately in order to change their passwords.

With One Identity Manager, you can eliminate these issues and significantly simplify password management in an enterprise environment that includes multiple data management systems.

One Identity Manager provides a cost-effective and efficient way to synchronize user passwords from an Active Directory domain to other data systems used in your organization. As a result, users can access other data management systems using their Active Directory domain password. Whenever a user password is changed in the source Active Directory domain, this change is immediately and automatically propagated to other data systems, so each user password remains in sync within the data systems at all times.

You must connect One Identity Manager to the data systems in which you want to synchronize passwords.

Related Topics

- Steps to Automate Password Synchronization on page 6
- Managing the Password Capture Agent on page 7
- Fine-Tuning Automated Password Synchronization on page 10

Steps to Automate Password Synchronization

NOTE: The Webservice has to be installed. For more information read the One Identity Manager Configuration Guide.

To automatically synchronize passwords from an Active Directory domain to another data system

1. Connect One Identity Manager to the Active Directory domain where you want to install the Password Capture Agent.
2. Connect One Identity Manager to the data system where you want to synchronize user account passwords with those in the source Active Directory domain.
3. Ensure that user accounts in the source Active Directory domain and the connected target systems are properly mapped to employees in One Identity Manager.

   For more information on how to assign employees to user accounts, see the One Identity Manager Administration Guide for Connecting to Active Directory.

4. Install the Password Capture Agent on each domain controller in the Active Directory domain you want to have as a source for password synchronization operations.

   The Password Capture Agent tracks changes to user passwords in the source Active Directory domain and provides this information to the Webservice (an optional component of One Identity Manager), which in turn synchronizes passwords in the target connected systems you specify.

After you have completed the above steps, the Password Capture Agent starts to automatically track user password changes in the source Active Directory domain and the One Identity Manager synchronizes passwords in the target connected system.

If necessary, you can fine-tune password synchronization settings by completing these optional tasks:

- Modify the default Password Capture Agent settings before installation.
- Modify the default Webservice settings related to password synchronization.
- Specify a custom certificate for encrypting the password synchronization traffic between the Password Capture Agent and the Webservice. By default, password synchronization traffic between the Password Capture Agent and the Webservice will be secured by transport layer security only.

Related Topics

- Managing the Password Capture Agent
- Configuring Password Capture Agent on page 11
- Specifying a Custom Certificate for Encrypting Password Synchronization Traffic on page 22

Managing the Password Capture Agent

The Password Capture Agent is required to track changes to user passwords in the Active Directory domain which you want to be the authoritative source for password synchronization operations. To synchronize passwords, you must install the One Identity Manager Password Capture Agent on each domain controller in the source Active Directory domain.

Whenever a password changes in the source Active Directory domain, the Password Capture Agent captures that change and sends the changed password securely to One Identity Manager. In turn, One Identity Manager uses the provided information to synchronize passwords in the target connected systems according to your settings.

Detailed information about this topic

- System Requirements for Password Capture Agent on page 8
- Installing the Password Capture Agent on page 8
- Using Windows PowerShell to Install the Password Capture Agent on page 8

- Uninstalling the Password Capture Agent on page 9
- Using Windows PowerShell to Uninstall the Password Capture Agent on page 10

System Requirements for Password Capture Agent

The following system prerequisites are the minimum requirements for installing and operating Password Capture Agent.

- Windows operating system

  Following versions are supported:

  - Windows Server 2008 R2 (non-Itanium based 64-bit) Service Pack 1 or later
  - Windows Server 2012
  - Windows Server 2012 R2
  - Windows Server 2016
- Microsoft .NET Framework Version 4.5.2 or later

Installing the Password Capture Agent

You can use this method to manually deploy the Password Capture Agent on each domain controller in the source Active Directory domain.

To manually install the Password Capture Agent

1. On a 64-bit domain controller, run the file `Password Capture Agent.msi`.
2. Step through the wizard to complete the Password Capture Agent installation.

Using Windows PowerShell to Install the Password Capture Agent

The Password Capture Agent provides a Windows PowerShell module for remote and automated installing, configuring and uninstalling. You can use this method to automatically deploy the Password Capture Agent on each domain controller in the source Active Directory domain.

For installing the Password Capture Agent remotely you should have prepared:

- the thumbprint of the certificate for password encryption, for example `1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188`
- the URL to the Webservice, for example `https://servername.domain.com/SoapService/Q1IMService.asmx`

Use the following commands in an elevated Windows PowerShell.

```
Import-Module OneIM-PasswordCaptureAgentMgmt
$ConfigProfile = New-PCAConfigProfile
$ConfigProfile['WebClient.WebServiceURL'].ConfigValue = '<Your URL>'
$ConfigProfile['WebClient.WebServiceType'].ConfigValue = 'Soap'
$ConfigProfile['WebClient.AuthenticationType'].ConfigValue = 'WindowsIntegrated'
$ConfigProfile['Backend.AuthenticationModule'].ConfigValue = 'DialogUser'
$ConfigProfile['Backend.Credential'].ConfigValue = Get-Credential viCaptureAgent
$ConfigProfile['Backend.CertificateThumbprint'].ConfigValue = '<Your Thumbprint>'
Install-PasswordCaptureAgent`
-ComputerName <Computer name>`
–LogFile <Full UNC path to the log file on the remote server>`
-LogVerbose`
-Setup <UNC path for Password Capture Agent MSI>`
-ConfigurationProfile $ConfigProfile
```

NOTE: To check that the Password Capture Agent is properly installed and working, you can examine the event viewer on the deployed server. The Password Capture Agent has its own log in the event viewer. The Password Capture Agent logs its summary status to this log after every system start and other such notable events during run-time.

Related Topics

- The Password Capture Agent Windows PowerShell Module on page 25
- Event Log for the Password Capture Agent on page 31

Uninstalling the Password Capture Agent

To remove Password Capture Agent open the list of installed programs on the computer on which the Active Roles is installed.

To remove Password Capture Agent using the control panel

1. Select **Programs and Features** in the Control Panel.
2. Double click on "One Identity Manager Password Capture Agent" in the list of installed programs.
3. Follow the on-screen instructions to uninstall the Password Capture Agent.

Using Windows PowerShell to Uninstall the Password Capture Agent

The Password Capture Agent provides a Windows PowerShell module for remote and automated installing, configuring and uninstalling. You can use this method to automatically deploy the Password Capture Agent on each domain controller in the source Active Directory domain.

For uninstalling the Password Capture Agent remotely use the following command in an elevated Active Directory.

```
Import-Module OneIM-PasswordCaptureAgentMgmt

Uninstall-PasswordCaptureAgent`

-ComputerName <Computer name>`

-LogFile <UNC path to log file>`

–LogVerbose
```

Related Topics

- The Password Capture Agent Windows PowerShell Module on page 25

Fine-Tuning Automated Password Synchronization

This section provides information about the optional tasks related to configuring automated password synchronization from an Active Directory domain to connected data systems.

Detailed information about this topic

- Configuring Password Capture Agent on page 11
- Configuring the Webservice on page 21
- Specifying a Custom Certificate for Encrypting Password Synchronization Traffic on page 22

Configuring Password Capture Agent

The Password Capture Agent has several settings you can modify. After you install the Password Capture Agent, each of these parameters is assigned a default value.

NOTE: If you do not configure the thumbprint for the Password Capture Agent, the password will be secured by transport layer security only (HTTPS).

Detailed information about this topic

- Configuration Parameters on page 11
- Secured Configuration Parameters on page 13
- Authentication Options on page 16
- Password on page 20
- Delete Jobs on page 20

Configuration Parameters

Some of the configuration parameters for the Password Capture Agent are changeable with the Windows Registry Editor. The parameters are split up into those that are used by the Password Capture Agent service and those used by the Password Capture Agent driver.

The base path for the parameters of the Password Capture Agent service is:

```
HKLM\SOFTWARE\One Identity\One Identity Manager\Password Capture Agent\Service\
```

Configuration parameters for the Password Capture Agent service

Table 1: parameter "WebService_URL"

Default	Type	Description
	String	This setting determines the location - Uniform Resource Locator (URL) - of the Webservice to which the Password Capture Agent provides information about changed user passwords. In the Form: https://<serverfqdn>/SoapService/Q1IMService.asmx

Table 2: parameter "CertificateThumbprint"

Default	Type	Description
	String	This setting specifies a certificate used to encrypt the data transfer channel between the Password Capture Agent and the Webservice. The certificate must be accessible both for the Password Capture

Default	Type	Description
		Agent and the Webservice. NOTE: If you disable this setting or do not configure it, the password will be secured by transport layer security only (HTTPS).

The base path for the parameters of the Password Capture Agent driver is:

`HKLM\SOFTWARE\One Identity\One Identity Manager\Password Capture Agent\Driver\`

NOTE: No reboot is required to take effect.

Configuration parameters for the Password Capture Agent driver

Table 3: parameter "Diagnostic"

Default	Type	Description
0	DWORD	Controls the logging behavior of Password Capture Agent driver. If enabled event log logging will be verbose, if the parameter "Logfile" has been set, additional trace logging will be written to that log file.

Table 4: parameter "FaultToleranceWaitTimeBeforeRetryInSeconds"

Default	Type	Description
120	DWORD	Time to wait in seconds before attempting a retry after a connection error.

Table 5: parameter "Logfile"

Default	Type	Description
	String	Diagnostics log file that should be used in addition to event log logging.

Table 6: Configuration parameter "LoggingSuccessfulOperations"

Default	Type	Description
0	DWORD	Enable to force the One Identity Manager to log successful transmissions to the Webservice to the event log.

Table 7: parameter "RequiredServices"

Default	Type	Description
RpcSs EventSystem COMSysApp	MultiString	Services that Password Capture Agent driver is waiting for, before starting the Password Capture Agent service.

Table 8: parameter "Ignoring\PasswordResetOperations"

Default	Type	Description
0	DWORD	Enable to force One Identity Manager to ignore password resets and only transmit password changes to One Identity Manager Service.

Table 9: parameter "Ignoring\UserNames"

Default	Type	Description
^.*$$	MultiString	Regular expressions that identify accounts that should be ignored. By default '^.*$$' ignores all machines accounts, e.g.: accounts ending with a $..

Table 10: parameter "Ignoring\UserRids"

Default	Type	Description
500 501 502	MultiString	UserRIDS that should be ignored by default. The default ignores built-in accounts.

Related Topics

- Advanced and Diagnostic Settings for the Password Capture Agent on page 33

Secured Configuration Parameters

The configuration parameters in this section are secured using the Microsoft Cryptography API and are not directly accessible. If you want to change or review these parameters after the Password Capture Agent installation use either the command line `Set-ServiceConfig.exe` or the Password Capture Agent Windows PowerShell module.

The command line will be supplied with the Password Capture Agent and is located in the Password Capture Agent installation folder `...\Service`.

Example (local)

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" WebServiceClientSkipHttpsValidation:0
```

NOTE: Retrieving secured configuration parameters requires a privileged user account. The process used to query for secured configuration parameters has to be elevated to retrieve parameter values.

Secured configuration parameters for Password Capture Agent

Table 11: parameter "WebServiceType"

Default	Allowed Values	Description
REST	REST | Soap	Specifies wether the Webservice at should be accessed using REST Api (AppServer) or Soap Api (SoapService).

Table 12: parameter "WebServiceClientSkipHttpsValidation"

Default	Allowed Values	Description
0	0 | 1	If enabled, HTTPS connections will be established without validation. This is potentially insecure and should never be used in production.

Table 13: parameter "WebServiceClientCredentialType"

Default	Allowed Values	Description
WindowsIntegrated	WindowsIntegrated | Certificate	Specifies if the authentication against the Internet Information Services (IIS) should use Windows integrated authentication or certificate based authentication.

Table 14: parameter "WebServiceClientCredentialCertificateFindByType"

Default	Allowed Values	Description
FindByThumbprint		Specifies how to search for the authentication certificate. All values of the `X509FindType`-Enumeration are allowed. Used in combination with "WebServiceClientCredentialType=Certificate".

Table 15: parameter "WebServiceClientCredentialCertificate"

Default	Allowed Values	Description
		Finds the certificate based on the find type defined in the configuration parameter "WebServiceClientCredentialCertificateFindByType". Used in combination with "WebServiceClientCredentialType=Certificate".

Table 16: parameter "BackendClientCredentialType"

Default	Allowed Values	Description
DialogUser	DialogUser | WebADS | ADSAccount	Specifies how to authenticate against One Identity Manager. "WebADS" and "ADSAccount" reuse the Windows credentials used for authentication against IIS. • ADSAccount = One Identity Manager 7.x • WebADS = One Identity Manager 6.1.x

Table 17: parameter "BackendClientCredentialUserName"

Default	Allowed Values	Description
viCaptureAgent		Specifies a system user for the authentication against One Identity Manager. Used in combination with "BackendClientCredentialType=DialogUser".

Table 18: parameter "BackendClientCredentialUserPwd_AcceptEmpty"

Default	Allowed Values	Description
0	0 | 1	Required if your system user is using a blank password. This is potentially insecure and should never be used in production. Used in combination with "BackendClientCredentialType=DialogUser" .

NOTE: The parameter "BackendClientCredentialUserPwd" is a write only parameter. The currently configured value cannot be retrieved using `Set-ServiceConfig`.

Example 1: Retrieve information about a secured configuration parameter

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" Describe:WebServiceClientCredentialType

Configuration parameter 'BackendClientCredentialType':

Name: BackendClientCredentialType

Possible values: DialogUser;WebADS;ADSAccount

Default value: DialogUser

Corresponding installer property: PROP_BACKEND_CLIENT_CREDENTIAL_TYPE

Description: Specify one of the credential types for authentication against the One Identity Manager
```

```
Present in installer GUI: Yes
Write only (read out not allowed): No
Read only (setting not allowed): No
Public in registry: No
Hint:
Comment:
```

Example 2: Retrieving a secured configuration parameter

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" Get:WebServiceClientCredentialType
WebServiceClientCredentialType=Certificate
Value was written to stderr.
Get configuration parameter - operation done.
```

Related Topics

- Authentication Options on page 16
- The Password Capture Agent Windows PowerShell Module on page 25

Authentication Options

The One Identity Manager Password Capture Agent supports several authentication options that can be configured separately for the authentication against the IIS hosting the Webservice and for the authentication against the One Identity Manager database.

Detailed information about this topic

- Authentication against the Webservice on page 16
- Authentication against One Identity Manager

Authentication against the Webservice

The authentication against the Webservice can be configured with the secured configuration parameter "WebServiceClientCredentialType".

Table 19: Options for parameter "WebServiceClientCredentialType"

Option	Description
WindowsIntegrated	This option uses the credentials of the user running the Password Capture Agent service to authenticate against the IIS hosting the

Option	Description
	Webservice. By default, this is the user "Local System" which uses the machine account to authenticate over the network. You can change the user of the Password Capture Agent service. The user requires administrative privileges to access the configuration parameters.
Certificate	This option uses a certificate to authenticate against the IIS hosting the Webservice. The certificates will be searched in `Cert: \CurrentUser\My\` and if not found in `Cert: \LocalMachine\My\`. Ensure that the user running the Password Capture Agent service has enough permissions to access the private key of the certificate.

Related Topics

- Secured Configuration Parameters on page 13
- Certificate Lookup Options on page 44

Authentication against One Identity Manager

The authentication against the One Identity Manager database can be configured with the secured configuration parameter "BackendClientCredentialType".

Table 20: Options for parameter "BackendClientCredentialType"

Option	Description
DialogUser	The One Identity Manager service uses the credentials stored in "BackendClientCredentialUserName" and "BackendClientCredentialPwd" to login as One Identity Manager system user. You can test your configuration by running the Object Browser with the system user login.
ADSAccount	This option uses the credentials of the user running the Password Capture Agent service to authenticate against the One Identity Manager database. This option is working for One Identity Manager version 7.x or later. NOTE: The user account has to be synchronized into by the One Identity Manager database and needs to be linked to an employee where the system user property is set accordingly. A machine account will not be able to authenticate against the One Identity Manager database. You can test your configuration by running the Object Browser with the same credentials as the Password Capture Agent service and using the Active Directory user account login.
WebADS	This option behaves the same as the option "ADSAccount" but is working for One Identity Manager version 6.1.x.

Example 1: Windows authentication and One Identity Manager system user login

The Password Capture Agent service uses Windows authentication to authenticate against the IIS with the Webservice running. To authenticate against One Identity Manager the system user "viCaptureAgent" is used.

- Prerequisites

  Configure the IIS site to only use Windows authentication for the Webservice.

- Testing

  You should be able to access the Webservice with a browser and the given WindowsActive Directory user account. Start a Windows PowerShell and try to access the Webservice using the given user account.

  ```
  Invoke-WebRequest -Uri https://servername.domain.com/SoapService/ -Credential
  $(Get-Credential <AD domain>\<AD user account>)
  ```

  You should be able to log into the Object Browser using the system user login and the credentials provided.

- Password Capture Agent configuration settings
  - WebServiceClientCredentialType = WindowsIntegrated
  - BackendClientCredentialType = DialogUser
  - BackendClientCredentialUserName = viCaptureAgent
  - BackendClientCredentialUserPwd = viCaptureAgentPasswordHere

Example 2: Windows authentication and Active Directory login

The Password Capture Agent service uses Windows authentication to authenticate against the IIS with the Webservice running. The Windows user account used to authenticate against the IIS will be reused to authentication against One Identity Manager.

- Prerequisites
  - Configure the IIS site to only use Windows authentication for the Webservice.
  - Configure IIS site to allow given users to access the Webservice (authorization).
  - The Password Capture Agent service is not allowed to run as "Local System" and requires an administrative user account to run with.
  - Given user accounts have to be known to the One Identity Manager database and have to be linked to an employee that has a system user configured to use for this type of authentication.
- Testing

  You should be able to access the Webservice with a browser and the given Active Directory user account. Start a Windows PowerShell and try to access the Webservice using the given user account.

```
Invoke-WebRequest -Uri https://servername.domain.com/SoapService/ -Credential
$(Get-Credential <ADDomain>\<ADUser>)
```

You can test your configuration by running the Object Browser as the given user account and using the Active Directory user account login.

- Password Capture Agent configuration settings
  - WebServiceClientCredentialType = WindowsIntegrated
  - BackendClientCredentialType = ADSAccount

Example 3: Certificate authentication and One Identity Manager system user login

This scenario allows you to connect from a host outside of your Active Directory domain. Stored credentials will be used to authenticate against One Identity Manager as system user.

- Prerequisites
  - Configure the IIS site to use HTTPS and Client Certificate Mapping. If you are not using Active Directory Certificate Services, you need to map the certificate to an Active Directory user account within IIS.
  - Client certificate with private key installed on the domain controller.
- Testing

  You should be able to access the Webservice with a browser using the given certificate. Start a Windows PowerShell as the user with the assigned certificate and try to access the Webservice.

  ```
  Invoke-WebRequest -Uri https://servername.domain.com/SoapService/ -
  CertificateThumbprint <ThumbprintOfGivenCertificate>
  ```

  You should be able to log into the Object Browser using the system user login and the credentials provided.
- Password Capture Agent configuration settings
  - WebServiceClientCredentialType = Certificate
  - WebServiceClientCredentialCertificateFindByType = FindByThumbprint
  - WebServiceClientCredentialCertificate = 0123456789ABCED0123456789ABCED0123456789
  - BackendClientCredentialType = DialogUser
  - BackendClientCredentialUserName = viCaptureAgent
  - BackendClientCredentialUserPwd = viCaptureAgentPasswordHere

Related Topics

- Secured Configuration Parameters on page 13

Password

To change the password used to authenticate against One Identity Manager use either the command line `Set-ServiceConfig.exe` or the Password Capture Agent Windows PowerShell module.

The command line will be supplied with the Password Capture Agent and is located in the Password Capture Agent installation folder `...\Service`.

NOTE: It is required that the Password Capture Agent is configured to use the parameter "BackendClientCredentialType = DialogUser".

Example (local)

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" BackendClientCredentialUserPwd:<new password>
```

The command line can also be used to set the password on a remote server on which the Password Capture Agent is installed. Use the optional parameter "Servername" to specify the name or the IP address of the remote server. In this case, COM+ Network Access must be enabled on the remote server in the application server role. If it is not enabled, see the Microsoft documentation to enable it (http://technet.microsoft.com/en-us/library/cc731967.aspx).

Example (remote)

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" BackendClientCredentialUserPwd:<new password> Servername: <Server name or IP address>.
```

NOTE: It is not required to restart the Password Capture Agent service. The new password takes effect immediately.

Related Topics

- The Password Capture Agent Windows PowerShell Module on page 25

Delete Jobs

The Password Capture Agent manages a queue with the password change jobs he is sending to One Identity Manager. If you need to delete some of these jobs from the internal queue you can use the command line `Set-ServiceConfig`.

Example (local)

```
"%ProgramFiles%\One Identity\One Identity Manager\Password Capture Agent\Service\Set-ServiceConfig.exe" <Job-ID>::=<YYYY.MM.DD HH.MM.SS.mmm>|*
```

Sample for a certain Job-ID: '2014.10.03 16:45:07.647'.

TIP: To delete all jobs use '*' as Job-ID.

Logging with NLog

Starting with Version 2.0, the Password Capture Agent is using NLog for logging. NLog allows the logging to be configured using an XML file.

By default we provide an `nlog.config` in the Password Capture Agent installation folder, which is using the same EventLog as previous Versions.

This `nlog.config` also provides additional examples on how configure NLog to log directly to a file or other tools such as chainsaw, you can enable these by uncommenting the matching rules in the rules section of the `nlog.config`.

More detailed examples, on how to configure NLog, can be found here:

https://github.com/nlog/NLog/wiki/Configuration-file

Be aware that a faulty `nlog.config` will cause the Password Capture Agent to stop logging.

Configuring the Webservice

You can modify the default values of the following configuration parameters related to password synchronization. You can modify these configuration parameters in the Designer.

Table 21: Parameters and default values

Parameter	Description
QER\Person\UseCentralPassword\ PasswordCaptureAgent\Certificate	This configuration parameter specifies if a certificate is used to encrypt the password synchronization traffic between the Password Capture Agent and the Webservice. Default value: enabled
QER\Person\UseCentralPassword\ PasswordCaptureAgent\SyncToSystemPassword	When this configuration parameter is set the Password Capture Agent synchronizes the Active Directory password to the employee's system password as well. Default value: enabled

IMPORTANT: Passwords for user accounts marked as privileged user accounts in the One Identity Manager will not be synchronized with other connected target systems.

TIP: If you have configured more than one Active Directory domain or have employees with more than one user account to use the Password Capture Agent check your password policy for employee's central password. To avoid circular password resets the password history value should be 1 or greater.

Specifying a Custom Certificate for Encrypting Password Synchronization Traffic

By default, the password synchronization traffic between the Password Capture Agent and the Webservice will be secured by transport layer security only. Therefore, it is strongly recommended that you to specify a custom certificate for this purpose.

IMPORTANT: You need a certificate file including the private key to encrypt the password synchronization traffic.

This section describes how to use a custom certificate for encrypting the password synchronization traffic.

Detailed information about this topic

- Step 1: Import Certificate into Certificates Store on page 22
- Step 2: Copy Certificate's Thumbprint on page 23
- Step 3: Provide Certificate's Thumbprint to the Password Capture Agent on page 24

Step 1: Import Certificate into Certificates Store

In this step, you import the certificate to the machine certificate store **Personal\Certificates** by using the Certificates snap-in. You must complete this step on each domain controller running the Password Capture Agent and on each computer running the Webservice that will participate in password synchronization.

To import the certificate

1. Open the Certificates - Local Computers snap-in.
2. In the console tree, click the logical store **Personal\Certificates**.
3. On the menu **Action**, point to **All Tasks** and then click **Import**.
4. Step through the wizard.
5. On the page "File to Import", in the text box **File name**, type the file name containing the certificate to be imported or click **Browse** and to locate and select the file. When finished, click **Next**.
6. On the page "Password", type the password used to encrypt the private key, and then click **Next**.

7. On the page "Certificate Store", ensure that the option **Place all certificates in the following store** is selected and the text box **Certificate store** displays "Personal", and then click **Next**.
8. On the page "Completion", revise the specified settings and click **Finish** to import the certificate and close the wizard.

To add read permissions to the certificate for the Webservice

1. Open the Certificates - Local Computers snap-in.
2. In the console tree, click the logical store **Personal\Certificates**.
3. Select your imported certificate from the list.
4. On the menu **Action**, point to **All Tasks** and then click **Manage Private Keys**.
5. Add "Read Permissions" for the security principal "NETWORK SERVICE" and click **Okay**.

Related Topics

- Step 2: Copy Certificate's Thumbprint on page 23
- Step 3: Provide Certificate's Thumbprint to the Password Capture Agent on page 24

Step 2: Copy Certificate's Thumbprint

In this step, you copy the thumbprint of your custom certificate. In the next step, you will need to provide the thumbprint to the Password Capture Agent.

To copy the thumbprint of your custom certificate

1. Open the Certificates - Local Computer snap-in.
2. In the console tree, click the store **Personal** to expand it.
3. Click the store **Certificates** to expand it.
4. In the details pane, double-click the certificate.
5. In the dialog box **Certificate**, click the tab **Details**, and scroll through the list of fields to select **Thumbprint**.
6. Copy the hexadecimal value of thumbprint to clipboard.

NOTE: You will need the copied thumbprint value to configure the Password Capture Agent.

Related Topics

- Step 1: Import Certificate into Certificates Store on page 22
- Step 3: Provide Certificate's Thumbprint to the Password Capture Agent on page 24

Step 3: Provide Certificate's Thumbprint to the Password Capture Agent

This step assumes that the Password Capture Agent Windows PowerShell module for the Password Capture Agent is installed on your workstation and all other requirements are met.

To provide the thumbprint to the Password Capture Agent

1. Sign on to the workstation installed with Password Capture Agent Windows PowerShell module as member of the group "Domain Admins".
2. Open an elevated command line.
3. Execute command to modify the configuration profile with the new thumbprint.

   ```
   REG ADD "\\<COMPUTERNAME>\HKLM\Software\One Identity\One Identity Manager\Password Capture Agent\Service" /v "CertificateThumbprint" /t REG_SZ /d "1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188"
   ```

4. Execute commands to restart the Password Capture Agent service.

   ```
   sc \\COMPUTERNAME stop "Password Capture Agent"
   ```

   ```
   sc \\COMPUTERNAME start "Password Capture Agent"
   ```

Related Topics

- Step 1: Import Certificate into Certificates Store on page 22
- Step 2: Copy Certificate's Thumbprint on page 23
- The Password Capture Agent Windows PowerShell Module on page 25

A

Appendix

The Password Capture Agent Windows PowerShell Module

The Password Capture Agent Windows PowerShell module was designed to simplify setup and management of Password Capture Agent on domain controllers. This module requires Windows PowerShell Remoting to be configured and enabled on the domain controllers to establish a connection and execute the commands.

This Windows PowerShell module is intended to be used on a Windows workstation with Windows PowerShell version 3.0 or later installed and whilst being logged on with a domain account that is in the built-in group "Domain Admins". The Password Capture Agent installer needs to be placed on a network share or copied manually to all domain controllers.

To allow administrators to better check for configuration errors we integrated some validations to our functions that will display warnings on any possible misconfiguration, for example, if the password encryption certificate is not installed.

NOTE: This module does not install the Webservice. This module does not generate and install the certificate required to encrypt passwords sent to the web service.

Prerequisites

The Password Capture Agent Windows PowerShell module has different requirements for the workstation or server the module is running on and for the domain controllers where the Password Capture Agent is installed.

Related Topics

- Executing the Password Capture Agent Windows PowerShell Module on page 26
- Configuration Targets on page 26

Executing the Password Capture Agent Windows PowerShell Module

The Password Capture Agent Windows PowerShell module requires Windows PowerShell version 3.0 or later. It is recommended to use Windows PowerShell version 4.0 if you are running Windows Server 2008 R2 or later.

The execution policy for Windows PowerShell has to allow the execution of signed scripts. For more information, see the execution policy guide for Windows PowerShell (http://technet.microsoft.com/en-us/library/hh847748.aspx).

Configuration Targets

To be able to use the Password Capture Agent Windows PowerShell module to remotely configure the Password Capture Agent on the domain controllers, these servers need to have Windows PowerShell Remoting configured and enabled. For more information, see the remote troubleshooting guide for Windows PowerShell (http://technet.microsoft.com/en-us/library/hh847850.aspx).

Installing the Password Capture Agent Windows PowerShell Module

To install the Password Capture Agent Windows PowerShell module

- Copy the folder `OneIM-PasswordCaptureAgentMgmt`, including content, to `C:\Windows\System32\WindowsPowerShell\v1.0\Modules\` - the systems global Password Capture Agent Windows PowerShell module path.

  - OR -

- Copy this folder `OneIM-PasswordCaptureAgentMgmt` to any path on your host, and add this path to the environment variable `PSModulePath`.

Before installing Password Capture Agent on a domain controller:

- Ensure that Webservice is installed and configured.
- Ensure that the certificate, to decrypt passwords with, is installed with private key in `LocalMachine\My` certificate store on the server hosting the Webservice.

- Ensure that the certificate, to encrypt passwords with, is installed with private key in `LocalMachine\My` certificate store on all domain controllers.

You should have prepared:

- the thumbprint of the certificate for password encryption, for example:

  `1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188`

- the URL to the Webservice, for example:

  `https://servername.domain.com/SoapService/Q1IMService.asmx`

Using the Password Capture Agent Windows PowerShell Module

Using the Password Capture Agent Windows PowerShell module to install Password Capture Agent on a specific domain controller

1. Sign on to the workstation installed with Password Capture Agent Windows PowerShell module as member of the group "Domain Admins".
2. Copy `Password Capture Agent.msi` to a network share that can be accessed by you on all domain controllers. e.g."\\StorageServer\SHARE\Password Capture Agent.msi".
3. Open an elevated Windows PowerShell.
4. Execute Command:

   ```
   Import-Module OneIM-PasswordCaptureAgentMgmt
   ```

5. Execute commands to define your configuration profile:

   ```
   $ConfigProfile = New-PCAConfigProfile

   $ConfigProfile['WebClient.WebServiceURL'].ConfigValue =
   'https://server.domain.com/SoapService/Q1IMService.asmx'

   $ConfigProfile['WebClient.WebServiceType'].ConfigValue = 'Soap'

   $ConfigProfile['Backend.Credential'].ConfigValue = Get-Credential viCaptureAgent

   $ConfigProfile['Backend.CertificateThumbprint'].ConfigValue =
   '1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188'
   ```

6. Execute Command:

   ```
   Install-PasswordCaptureAgent`

   -ComputerName "DC01.DEMOCORP.COM"`

   -Setup "\\StorageServer\SHARE\Password Capture Agent.msi"`

   -ConfigurationProfile $ConfigProfile
   ```

   By running this command, you install the Password Capture Agent on `DC01.DEMOCORP.COM`. The installation will be run off a network location and WebServiceURL/CertificateThumbprint are passed to the setup.

Because the `-Restart` switch is not specified, the domain controllers will not automatically reboot after successful installation.

Using the Password Capture Agent Windows PowerShell module to install Password Capture Agent on all domain controllers

1. Sign on to workstation with installed Password Capture Agent Windows PowerShell module as member of the group "Domain Admins".
2. Copy `Password Capture Agent.msi` to a network share that can be accessed by you on all domain controllers. e.g."\\StorageServer\SHARE\Password Capture Agent.msi".
3. Open an elevated Windows PowerShell.
4. Execute Command:

   ```
   Import-Module OneIM-PasswordCaptureAgentMgmt
   ```

5. Execute commands to define your configuration profile:

   ```
   $ConfigProfile = New-PCAConfigProfile

   $ConfigProfile['WebClient.WebServiceURL'].ConfigValue =
   'https://server.domain.com/SoapService/Q1IMService.asmx'

   $ConfigProfile['WebClient.WebServiceType'].ConfigValue = 'Soap'

   $ConfigProfile['Backend.Credential'].ConfigValue = Get-Credential viCaptureAgent

   $ConfigProfile['Backend.CertificateThumbprint'].ConfigValue =
   '1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188'
   ```

6. Execute Command:

   ```
   Get-DomainController | Install-PasswordCaptureAgent`

   -Setup \\StorageServer\SHARE\ One Identity Manager Password Capture Agent.msi`

   -ConfigurationProfile $ConfigProfile

   -Restart
   ```

   By running this command, you receive a list of domain controllers and sequentially start the install on each one. The install will be run off a network location and WebServiceURL/CertificateThumbprint are passed to the setup.

   Because the `-Restart` switch is specified, the domain controllers will automatically reboot after successful installation.

Working with Configuration Profiles

The Password Capture Agent Windows PowerShell module includes functions to create, show, get, set, import and export a Password Capture Agent configuration profile.

NOTE: The function `Show-PCAConfigProfile` may also be used to get an overview of all parameters and read their descriptions or destinations.

Getting and setting the configuration profile is only possible if the Password Capture Agent is installed and running. It is not possible to access the secured configuration parameters without it.

Example 1: Creating new profile and editing it

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = New-PCAConfigProfile

$ConfigProfile['WebClient.WebServiceURL'].ConfigValue =
'https://fqdn.democorp.com/Q1IMService/Q1IMService.asmx'

$ConfigProfile['WebClient.AuthenticationType'].ConfigValue = 'WindowsIntegrated'

$ConfigProfile['Backend.AuthenticationModule'].ConfigValue = 'DialogUser'

$ConfigProfile['Backend.Credential'].ConfigValue = Get-Credential viCaptureAgent

$ConfigProfile['Backend.CertificateThumbprint'].ConfigValue =
'0123456789ABCED0123456789ABCED0123456789'
```

Example 2: Read current profile and show it using GUI

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = Get-PCAConfigProfile

Show-PCAConfigProfile -ConfigurationProfile $ConfigProfile
```

Example 3: Read current profile and export it to xml

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = Get-PCAConfigProfile

Export-PCAConfigProfile -ConfigurationProfile $ConfigProfile -FilePath
C:\tmp\CurrentPCAConfig.xml
```

Example 4: Import profile, edit and set it

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = Import-PCAConfigProfile -Filepath C:\tmp\CurrentPCAConfig.xml

$ConfigProfile['Backend.CertificateThumbprint'].ConfigValue =
'0123456789ABCED0123456789ABCED0123456780'

Set-PCAConfigProfile -ConfigurationProfile $ConfigProfile
```

Example 5: Import profile and install Password Capture Agent

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = Import-PCAConfigProfile -Filepath C:\CurrentPCAConfig.xml

Install-PasswordCaptureAgent`

–LogFile <Full UNC path to the log file on the remote server>`

-Setup <UNC path for Password Capture Agent MSI>`

-ConfigurationProfile $ConfigProfile
```

Example 6: Changing parts of the configuration

```
Import-Module OneIM-PasswordCaptureAgentMgmt

$ConfigProfile = Get-PCAConfigProfile

$ConfigProfile['Backend.Credential'].ConfigValue = Get-Credential viCaptureAgent

Set-PCAConfigProfile -ConfigurationProfile $ConfigProfile
```

Example 7: Changing parts of the configuration on all domain controllers

```
Get-DomainController | Foreach-Object {
      $ConfigurationProfile = Get-PCAConfigProfile -ComputerName $_
      $ConfigurationProfile['Backend.CertificateThumbprint'].ConfigValue =
      '1800b62e8cf19d1c4bcdcd2b6e435c3c85e04188'
      Set-PCAConfigProfile -ComputerName $_ -ConfigurationProfile $ConfigurationProfile
      -RestartService
}
```

Troubleshooting

I am unable to import the Password Capture Agent Windows PowerShell module.

Windows PowerShell has an execution policy to restrict what may run. For more information about troubleshooting, see http://technet.microsoft.com/en-us/library/hh847850.aspx.

- Is the folder "OneIM-PasswordCaptureAgentMgmt" in any folder listed in `$env:PSModulePath`?

I am unable to establish a connection to the domain controllers.

The connection to the domain controllers requires Windows PowerShell Remoting to be configured and enabled. Also the firewall may block this connection by default. For more information about troubleshooting, see http://technet.microsoft.com/en-us/library/hh847850.aspx.

I am experiencing problems installing the Password Capture Agent. Is there a way to get a log file?

Yes. Both `Install-PasswordCaptureAgent` and `Uninstall-PasswordCaptureAgent` have parameters that allow you to specify a log file and if logging should be verbose. The log file will be used by `msiexec.exe`.

Example 1

```
Uninstall-PasswordCaptureAgent`
-ComputerName "DC01.DEMOCORP.COM"
```

```
-LogFile \\StorageServer\SHARE\DC01.uninstall.log`
-LogVerbose
```

Example 2

```
Install-PasswordCaptureAgent`
-ComputerName "DC01.DEMOCORP.COM"`
-LogFile \\StorageServer\SHARE\DC01.install.log`
-LogVerbose`
-Setup "\\StorageServer\SHARE\PasswordCaptureAgent.msi"
```

Is it possible to automatically reboot the domain controllers after installing / uninstalling Password Capture Agent?

Yes. Both Both `Install-PasswordCaptureAgent` and `Uninstall-PasswordCaptureAgent` have a switch called restart which will do exactly this. It is `$False` by Default.

Example 1

```
Uninstall-PasswordCaptureAgent -ComputerName "DC01.DEMOCORP.COM" -Reboot
```

Example 2

```
Uninstall-PasswordCaptureAgent -ComputerName "DC01.DEMOCORP.COM" -Reboot:$True
```

Advanced Scenarios and More Examples

With the Password Capture Agent Windows PowerShell module, there are many ways to install Password Capture Agent on your domain controllers. Use the built-in Windows PowerShell® help to find more examples of usage:

```
Get-Help Get-PasswordCaptureAgentServiceConfig -Full
Get-Help Set-PasswordCaptureAgentServiceConfig -Full
Get-Help Install-PasswordCaptureAgent -Full
Get-Help Uninstall-PasswordCaptureAgent -Full
```

Event Log for the Password Capture Agent

You can read the log Password Capture Agent in the event viewer, in the folder "Applications and Services Logs". It shows you details of hints, warnings and errors if they

occur.

- Level
- Date and time
- Source
- Event ID
- Track category

In addition, you find information about the configuration summary on every startup process.

Example

Configuration summary:

- This DLL: "C:\WINDOWS\system32\PCA_Driver.DLL"
- File Version: "1.0.1.9"
- DLL File Version: "1.0.1.9"
- Used log in event log: "One Identity Manager Password Capture Agent", with source name: 'Driver'
- Configuration key: "HKEY_LOCAL_Machine\SOFTWARE\One Identity\One Identity Manager\Password Capture Agent\Driver"
- Diagnostic mode: No
- Diagnostic beep frequency: Beep Off
- Deactivate on start: No
- Retry on error after seconds: 120
- Storage time of pending captures in days: 7
- Log file: "<no log file specified>"
- Domain name for accounts: "democorp"
- Companion service: "One Identity Manager Password Capture Agent" has successfully initialized
- Number of unfinished captures in queue: 0
- Driver initialization completed.

Related Topics

- Advanced and Diagnostic Settings for the Password Capture Agent on page 33

Customizing Security for the Password Capture Agent Service

You can limit the scope of users and groups that are permitted to configure the Password Capture Agent using built-in Windows techniques.

Use the COM+ Management Console to specify permissions for the task `SetConfigParameter` under "Component Services\Computers\My Computer\COM+ Applications\One Identity Manager Password Capture Agent\Components\PCA.Com_Class\Interfaces\COM_Interface\Methods".

Advanced and Diagnostic Settings for the Password Capture Agent

The Password Capture Agent offers several registry settings for diagnostic and special purposes. These parameters can be set in the registry. You will find them in the Registry Editor underneath the path `HKEY_LOCAL_MACHINE\SOFTWARE\One Identity\One Identity Manager\Password Capture Agent\Driver`.

NOTE: No reboot is required to take effect.

Detailed information about this topic

DeactivateOnStart

- Type: REG_DWORD
- Unit: switch (on/off)

- Default: 0 (= off)

Disable password change without uninstalling. If the value is set to 1 the Password Capture Agent will be disabled after the next reboot. The only action after reboot is a single hint, logged to the Password Capture Agent Event Log - named One Identity Manager Password Capture Agent - in the Windows Event Viewer.

Diagnostic

- Type: REG_DWORD
- Unit: Switch (on/off)
- Default: 0 (= off)

Enables some diagnostic behavior if this parameter is set to 1.

- Verbose logging to log file if it is specified (Registry Parameter `LogFile`). Every operation and its result will be logged.
- All logging will also be sent as an operating system debug message for appropriate live viewers (e.g. DebugView from Windows Sysinternals).
- In Windows Server every message to the Password Capture Agent event log in Windows Event Viewer will be additionally noted with a beep tone. Where an error message will be noted with a deep Tone: 200 Hz, a warning message will be noted with a middle-high tone: 1000 Hz, an info message will be noted with a high tone: 2000 Hz. This is only available in Windows Server 2003 service pack 2.
- The registry parameter `DiagnosticBeepFrequency` is enabled.
- The registry parameter `LogFile` is enabled.

Related Topics

- DiagnosticBeepFrequency on page 34
- LogFile on page 35

DiagnosticBeepFrequency

- Type: REG_DWORD
- Unit: tone frequency in Hz
- Default: 0

In Windows Server if is parameter `Diagnostic` is enabled, every diagnostic log message will be noted with a beep tone in the specified frequency. This is only available in Windows Server 2003 service pack 2.

Related Topics

- Diagnostic on page 34

FaultToleranceWaitTimeBeforeRetryInSeconds

- Type: REG_DWORD
- Unit: Seconds
- Default: 120

If an error occurred, the value specified is the wait time in seconds before a retry is executed. If the value is 0, a retry is immediately executed.

LogFile

- Type: REG_SZ
- Unit: File
- Default: <EMPTY>

Specifies a name for a log file which has to be created. If no value is specified, no log file is created. Only the file name without a path has to be specified, so the file will reside in the installation folder "%ProgramData%\One Identity\One Identity Manager\Password Capture Agent\Driver".

The log file logs all activities and the more details if parameter `Diagnostic` is enabled. The log file is read-only but can be accessed from any text viewer. It is always recreated on reboot and does not yet contain any history. The time format of the logged time stamps depends on the local language of the operating system and not on the user.

Related Topics

- Diagnostic on page 34

PendingCapturesArchiveDepthInDays

- Type: REG_DWORD
- Unit: Days
- Default: 7

Specifies the number of days for undelivered password changes to be saved for retrying. Undelivered password changes can arise if errors have occurred, for example, if the associated Webservice is not available due to network errors, time outs, and so on. Every password change that cannot be delivered is also logged to the Password Capture Agent event log in Windows Event Viewer. If 0 is specified, no undelivered password changes are saved; they will be lost.

Synchronous

- Type: REG_DWORD
- Unit: Switch (on/off)
- Default: 0 (= off)

If this parameter is set with a value of 1, every password change is handled sequentially, as a result, the initiating process will be blocked until all other components in the beyond processing chain have completed. Also all password change events occurring in parallel will be blocked until the current password change is completed. This setting also means that a user, who just changes his password in the password-change-dialog, must wait until the whole processing is completed. This setting is only for test purposes.

Ignoring\UserNames

- Type: REG_MULTI_SZ
- Unit: List of strings
- Default: "^.*$$"

This parameter specifies a list of names of accounts that are to be ignored and whose password changes are irrelevant and are not be tracked. This can be built-in accounts as the machine account and the guest account or other operating system related accounts as virtual machine accounts and the like. Every account in this list is specified as a regular expression. The default is the machine account ("^.*$$") which is to be ignored if changing its password.

Ignoring\UserRids

- Type: REG_MULTI_SZ
- Unit: List of numbers
- Default: 500, 501, 502

Specifies a list of RIDs of accounts (relative part of a user SID number) that are to be ignored and whose password changes are irrelevant and are not to be tracked. This are built-in accounts as the machine account, the guest account and the like. Every account in

this list is specified as an User-RID. RIDs of built-in accounts are the same on every machine. The default for this parameter is the RID of the built-in administrator account (500), the RID of the built-in guest account (501) and the RID of the built-in Kerberos ticket-granting-ticket account (502).

Achieving High Availability for the Webservice with Windows Network Load Balancing

This appendix describes how to achieve high availability for the Webservice using Network Load Balancing service.

The Network Load Balancing cluster requires a dedicated IP address and fully qualified domain name. This should be setup before installing the cluster. This fully qualified domain name will be used later to access the Webservice. This means, that every host needs a certificate that is valid for the chosen fully qualified domain name and trusted by each domain controller.

Hosts in a Network Load Balancing cluster require at least two network interface cards. The first network interface cards should be for general communication and maintenance and the second network interface cards should be dedicated to Network Load Balancing traffic.

To allow high availability in a Network Load Balancing cluster, you need multiple hosts installed and configured with Webservice. These hosts should be dedicated to that task. Installing Network Load Balancing on domain controllers is not supported.

Example settings in this lab with network interface card (NIC) and fully qualified domain name (FQDN):

Host1

- Web01.democorp.com (Windows Server 2012 R2)
  - NIC1: 192.168.0.20
  - NIC2: 192.168.0.200 (STATIC)

Host2

- Web02.democorp.com (Windows Server 2012 R2)
  - NIC1: 192.168.0.21
  - NIC2: 192.168.0.201 (STATIC)
- Network Load Balancing Cluster:
  - FQDN: ServiceCluster.democorp.com
  - IP: 192.168.0.50

Detailed information about this topic

- Step 1: Install the Windows Network Load Balancing Service on page 38
- Step 2: Configure Windows Network Load Balancing on page 38
- Step 3: Configuration Validation on page 40
- Step 4: Applying Password Capture Agent Web Service URL on the Password Capture Agent on page 40
- Troubleshooting on page 40

Step 1: Install the Windows Network Load Balancing Service

This step shows you how to install the required Windows feature to allow the configuration of Network Load Balancing. You should complete this task on all hosts that are supposed to be part of this cluster before continuing with the next step.

To install the required Windows feature (manually)

1. Start the Server Manager.
2. Click on **Add roles and Features**.
3. Skip the first page of the wizard.
4. Select **Role-based or feature-based installation**.
5. Select the server on which you want to install Network Load Balancing feature.
6. Click **next** on the server roles page.
7. Check **Network Load Balancing** on the features page.
8. Click **Add-Feature** on the menu.
9. Click **next** on the features page.
10. Click **install** on the confirmation page.

To install the required Windows feature (with Windows PowerShell)

1. Start a Windows PowerShell as administrator.
2. Enter `Install-Windows Feature NLB`.

Step 2: Configure Windows Network Load Balancing

This step shows you how to configure the Network Load Balancing process. This task will be executed on one of the hosts that should be clustered for Network Load Balancing. These

settings require you to have administrative privileges on the selected hosts.

To configure Network Load Balancing (manually)

1. Start Network Load Balancing Manager.
2. In the menu **Cluster**, click on **New**.
3. Preform the following tasks in the window "New Cluster: Connect".
    a. Connect to your first host, e.g.: web01.democorp.com and click **Connect**.
    b. In the list of network interfaces, select Ethernet 2 - with the IP that is dedicated to Network Load Balancing and set to "static".
    c. Click **Next**.
4. In the window "New Cluster: Host Parameters", click **Next**.
5. Preform the following tasks in the window "New Cluster: Cluster IP Addresses".
    a. Click **Add** and enter the Cluster IP: 192.168.0.50 with matching subnet mask.
    b. Click **Next**.
6. Preform the following tasks in the window "New Cluster: Cluster Parameters".
    a. Enter the Full Internet Name, e.g.: ServiceCluster.democorp.com.
    b. Click **Next**.
7. Preform the following tasks in the window "New Cluster: Port Rules".
    a. Select the existing rule and click **Remove**.
    b. Click **Add**.
8. Preform the following tasks in the window "Add/Edit Port Rule".
    a. Set the **Port range** to: From 443 To 443.
    b. Select "TCP" as protocol.
    c. Set the **Filtering Mode** to "Multiple Host".
    d. Set the **Affinity** to match your requirements or leave it at "Single (*)".
    e. Click **OK**.
    f. Click **Finish**.

(*) The affinity is used to determine to which back-end server a client is connected. The Webservice uses a stateless architecture, thus any affinity will work.

To add additional hosts to the Network Load Balancing cluster

1. Start Network Load Balancing Manager.
2. In the menu **Cluster**, click on **Connect to existing**.
3. In the window "Connect to Existing: Connect", enter the Cluster IP / FQDN and click **Connect**.
4. In the Clusters list, select the Cluster and click **Finish**.
5. In the tree view, select the cluster.

6. In the menu **Cluster**, click on **Add Host**.
7. Preform the following tasks in the window "Add Host to Cluster: Connect".
    a. Connect to your next host, e.g.: web02.democorp.com and click **Connect**.
    b. In the list of network interfaces, select Ethernet 2 - with the IP that is dedicated to Network Load Balancing and set to "static".
    c. Click **Next**.
8. In the window "Add Host to Cluster: Host Parameters", click **Next**.
9. In the window "Add Host to Cluster: Port Rules", click **Finish**.

Step 3: Configuration Validation

Before changing the configuration of the One Identity Manager Password Capture Agent, you must validate the configuration. After the previous steps, you should be able to access `https://ServiceCluster.democorp.com` and see the IIS welcome screen.

Step 4: Applying Password Capture Agent Web Service URL on the Password Capture Agent

To set the Password Capture Agent web service URL

1. Start an elevated command line.
2. Execute command to modify the web service URL at the Password Capture Agent.

```
REG ADD "\\<COMPUTERNAME>\HKLM\Software\One Identity\One Identity
Manager\Password Capture Agent" /v "WebService_URL" /t REG_SZ /d
"https://ServiceCluster.democorp.com/SoapService/Q1IMService.asmx"
```

3. Execute commands to restart the Password Capture Agent service.

```
sc \\<COMPUTERNAME> stop "Password Capture Agent"
```

```
sc \\<COMPUTERNAME> start "Password Capture Agent"
```

Troubleshooting

When accessing https://ServiceCluster.democorp.com I receive an invalid certificate error in my browser.

Since you are not accessing each host by its real host name, you have to ensure that the SSL certificate was issued to the common name matching the cluster's fully qualified

domain name and that the fully qualified domain name is set in the **Subject Alternative Names** (SAN) field.

When accessing https://ServiceCluster.democorp.com Kerberos authentication fails.

Since you are accessing all servers in this cluster with the same fully qualified domain name, Kerberos authentication will fail. If you have NT Lan Manager disabled as fallback, authentication will not work.

Installing the Password Capture Agent with MSIEXEC

The Password Capture Agent Setup can be automated using MSIEXEC parameters. The parameters are listed in the following table.

Parameters for MSIEXEC

Table 22: Parameter "PROP_WEBSERVICE"

Configuration after Setup	Values	Comment
Registry value: Service\WebService_URL		The Webservice URL.

Table 23: Parameter "PROP_WEB_SERVICE_TYPE"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: WebServiceType	REST | Soap	WebService Api Type.

Table 24: Parameter "PROP_CERTIFICATE"

Configuration after Setup	Values	Comment
Registry value: Service\CertificateThumbprint		The One Identity Manager password encryption certificate.

Table 25: Parameter "PROP_LOGGING_SUCCESSFUL_OPERATIONS"

Configuration after Setup	Values	Comment
Registry value: Driver\LoggingSuccessfulOperations	0 | 1 Default: 0	

Table 26: Parameter "PROP_IGNORE_PASSWORD_RESET_OPERATIONS"

Configuration after Setup	Values	Comment
Registry value: Driver\Ignoring\PasswordResetOperations	0 | 1 Default value: 0	

Table 27: Parameter "PROP_BACKEND_CLIENT_CREDENTIAL_TYPE"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: BackendClientCredentialType	DialogUser | WebADS | ADSAccount Default: DialogUser	

Table 28: Parameter "PROP_BACKEND_CLIENT_CREDENTIAL_USER_NAME"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: BackendClientCredentialUserName	Default: viCaptureAgent	

Table 29: Parameter "PROP_BACKEND_CLIENT_CREDENTIAL_USER_PWD"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: BackendClientCredentialUserPwd		

Table 30: Parameter "PROP_BACKEND_CLIENT_CREDENTIAL_USER_PWD_ACCEPT_EMPTY"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: BackendClientCredentialUserPwd_AcceptEmpty	0 | 1 Default: 0	

Table 31: Parameter "PROP_WEB_SERVICE_CLIENT_SKIP_HTTPS_VALIDATION"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: WebServiceClientSkipHttpsValidation	0 | 1 Default: 0	

Table 32: Parameter "PROP_WEB_SERVICE_CLIENT_CREDENTIAL_TYPE"

Configuration after Setup	Values	Comment
Set-ServiceConfig.exe: WebServiceClientCredentialType	WindowsIntegrated | Certificate Default: WindowsIntegrated	

Table 33: Parameter "PROP_WEB_SERVICE_CLIENT_CREDENTIAL_ CERTIFICATE_FIND_BY_TYPE"

Configuration after Setup	Values	Comment
Set-ServiceConfig.Exe: WebServiceClientCredentialCertificateFindByType	Default: FindByThum-bprint	

Table 34: Parameter "PROP_WEB_SERVICE_CLIENT_CREDENTIAL_ CERTIFICATE"

Configuration after Setup	Values	Comment
Set-ServiceConfig.Exe: WebServiceClientCredentialCertificate		

Table 35: Parameter "PROP_FINAL_FUNCTION_TEST"

Configuration after Setup	Values	Comment
Only used by - and during the setup.	0 | 1 Default: 1	Only used by setup to determine whether final function test should be executed. Failure will cause setup to fail.

NOTE: MSIEXEC does not recognize 0 to uncheck checkboxes, instead use `PROP_FINAL_FUNCTION_TEST=""` for example.

Example 1: Silent install with default settings

```
msiexec.exe /i "<SETUP_MSI_FILE>" /quiet /norestart /L "<LOGFILE>"
```

Example 2: Silent install with parameters

```
msiexec.exe /i "<SETUP_MSI_FILE>" /quiet /norestart PROP_WEBSERVICE="<WEBSERVICE_URL>"
PROP_WEBSERVICE_TYPE="<WEBSERVICE_TYPE>" PROP_CERTIFICATE="<CERTIFICATE_THUMBPRINT>"
PROP_BACKEND_CLIENT_CREDENTIAL_USER_NAME="<One Identity Manager system user>" PROP_
BACKEND_CLIENT_CREDENTIAL_USER_PWD="<System user password>" PROP_DENY_SELF_SIGNED_
CERTIFICATES_FOR_HTTPS="1" PROP_FINAL_FUNCTION_TEST="1" PROP_IGNORE_PASSWORD_RESET_
OPERATIONS="" /L "<LOGFILE>"
```

Example 3: Interactive Installation

```
msiexec.exe /i "<SETUP_MSI_FILE>" /norestart PROP_WEBSERVICE="<WEBSERVICE_URL>" PROP_
WEBSERVICE_TYPE="<WEBSERVICE_TYPE>" PROP_CERTIFICATE="<CERTIFICATE_THUMBPRINT>" PROP_
USERNAME="<One Identity Manager system user>" PROP_BACKEND_CLIENT_CREDENTIAL_USER_
PWD="<System user password>" PROP_DENY_SELF_SIGNED_CERTIFICATES_FOR_HTTPS="1" PROP_
FINAL_FUNCTION_TEST="1" PROP_IGNORE_PASSWORD_RESET_OPERATIONS="" /L "<LOGFILE>"
```

Example 4: Uninstall

```
msiexec.exe /X{E7D3E2C0-0BD9-4EBB-A70C-E835D575611B} /quiet /norestart /L "<LOGFILE>"
```

Certificate Lookup Options

Because certificates have a limited lifetime and therefore have to be renewed or updated, Password Capture Agent service has the option to configure the search for valid certificates. Be aware that not all configurable `FindByTypes` may be suitable for you needs.

Example 1: Use certificate from local trusted root certificate authority (Active Directory Certificate Services)

All certificates issued by "DEMOCORP DEMO ROOT CA" to be valid for this purpose. Automatically enrollment is used to distribute the certificates and new certificates will automatically be generated before expiration.

- WebServiceClientCredentialCertificateFindByType = FindByIssuerName
- WebServiceClientCredentialCertificate = "DEMOCORP DEMO ROOT CA"

- OR-

- WebServiceClientCredentialCertificateFindByType = FindByIssuerDistinguishedName
- WebServiceClientCredentialCertificate = "CN=DEMOCORP DEMO ROOT CA, DC=Democorp, DC=com"

Example 2: Use certificate based on subject

All certificates with a subject "demoadmn" to be valid for this purpose.

- WebServiceClientCredentialCertificateFindByType = FindBySubjectName
- WebServiceClientCredentialCertificate = "demoadmn"

- OR-

- WebServiceClientCredentialCertificateFindByType = FindBySubjectDistinguishedName
- WebServiceClientCredentialCertificate = "CN=demoadmn, CN=Users, DC=Democorp, DC=com"

Example 3: Use static certificate by thumbprint and change manually when new certificate is available

- WebServiceClientCredentialCertificateFindByType = FindByIssuerName
- WebServiceClientCredentialCertificate = 0123456789ABCED0123456789ABCED0123456789

Known Error Codes

There are several known error codes that the script `VI_CaptureAgent_SetPassword` can use to reject a password change. The script is stored in the Password Capture Agent database. If you feel that it does not suits your needs, you are able to overwrite the script.

Following is the list of possible errors and appropriate actions that are returned by the script `VI_CaptureAgent_SetPassword`.

Table 36: Errors and appropriate actions

Error Code	Error Message	Action	Adminstration Action
0	No Error. Change went through.	OK	-
1	Password cycle detected.	Skip	Check manual for password cycles.
2	ADS Account is marked as privileged and will not be handled.	Skip	-
1212	ADS Account has no domain.	Skip	-
1317	ADS Account is not known by One Identity Manager.	Skip	Check if your Active Directory domain has been configured to be synchronized regularly within One Identity Manager.
1332	ADS Account exists but is not mapped to a Person in One Identity Manager.	Skip	Check One Identity Manager configuration, you should not have Active Directory user accounts without mapped employees.
1355	ADS Domain is not known by One Identity Manager.	Skip	Check if your Active Directory domain has been configured to be synchronized within One Identity Manager.
9901	More than one ADS Account found in One Identity Manager database matching DOMAIN\SAMAccountName.	Skip	Check for duplicate entries in table `ADSAccount` within One Identity Manager.
9902	Failed to load Person mapped to	Skip	Check One Identity Manager for

Error Code	Error Message	Action	Adminstration Action
	ADS Account from One Identity Manager database.		problems, try loading that employee within Object Browser.
8205	Password encryption does not match the configuration in One Identity Manager.	Skip	Compare configuration of One Identity Manager and Password Capture Agent.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product