



leostream[®]

Remote Desktop Access Platform

Using RADIUS with the Leostream[®] Platform

Supporting Multi-Factor Authentication in your Leostream Environment

Contacting Leostream

Leostream Corporation
77 Sleeper St.
PMB 02-123
Boston, MA 02210
USA

<http://www.leostream.com>
Telephone: +1 781 890 2019

To submit an enhancement request, email features@leostream.com.

To request product information or inquire about our future directions, email sales@leostream.com.

Copyright

© Copyright 2002-2024 by Leostream Corporation

This software program and documentation are copyrighted by Leostream. The software described in this document is provided under a license agreement and may be used or copied only under the terms of this agreement. No part of this manual may be copied or reproduced in any form without prior written consent from Leostream.

Trademarks

The following are trademarks or registered trademarks of Leostream Corporation.

Leostream®

The Leostream graphical logo™

The absence of a product name or logo from this list does not constitute a waiver of the trademark or other intellectual property rights concerning that product, name, or logo by Leostream.

Microsoft, Active Directory, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Okta is a trademark of Okta, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders. Leostream claims no right to use of these marks.

Patents

Leostream software is protected by U.S. Patent 8,417,796.

Contents

CONTENTS	3
OVERVIEW	4
AUTHENTICATION WORKFLOW AND REQUIREMENTS	4
ENTERING A ONE-TIME PASSCODE.....	4
REQUESTING A PUSH NOTIFICATION.....	6
CONFIGURING THE CONNECTION BROKER TO COMMUNICATE WITH RADIUS SERVERS	7
TESTING AND TROUBLESHOOTING	8
SPECIFYING LEOSTREAM USERS WHO REQUIRE MFA.....	9
END-USER LOGIN WORKFLOW	10
EXAMPLE USING THE LEOSTREAM WEB CLIENT	10
EXAMPLE USING LEOSTREAM CONNECT.....	11
CUSTOMIZING THE LOGIN DIALOGS	12
EXAMPLE CONFIGURATION: OKTA	13
EXAMPLE CONFIGURATION: DUO	14
EXAMPLE CONFIGURATION: AZURE AD MFA AND MICROSOFT NPS	16
<i>Step 1: Create a new RADIUS client.....</i>	<i>16</i>
<i>Step 2: Add a new Network Policy.....</i>	<i>18</i>
<i>Step 3: Add a Connection Request Policy.....</i>	<i>20</i>
<i>Step 4: Install and configure NPS Extension for Azure MFA</i>	<i>22</i>
<i>Step 5: Add a RADIUS MFA provider to the Connection Broker</i>	<i>23</i>

Overview

The Leostream® Connection Broker can communicate with RADIUS servers to enable multi-factor authentication (MFA) for your end-user logins. Any RADIUS server or Identity Provider with a RADIUS component or agent, such as Okta and Duo, can be used with the Connection Broker.

RADIUS MFA is supported when users log into your Leostream environment using any of the following client devices.

- The Leostream Web client
- Leostream Connect for Windows
- Leostream Connect for Linux and macOS
- PCoIP Zero clients
- PCoIP Software clients

Users can perform MFA by entering a one-time passcode or requesting a push notification.

Authentication Workflow and Requirements

When using a RADIUS server to provide MFA for Leostream users, the first authentication factor is always username and password. You can use Active Directory, an OpenLDAP server, or even your Connection Broker as the first authentication server.

You then enable MFA for groups of users in your authentication server or per-user if the user is defined locally in the Connection Broker.

The authentication workflow depends on if the user enters a one-time passcode or requests a push notification.

Entering a One-Time Passcode

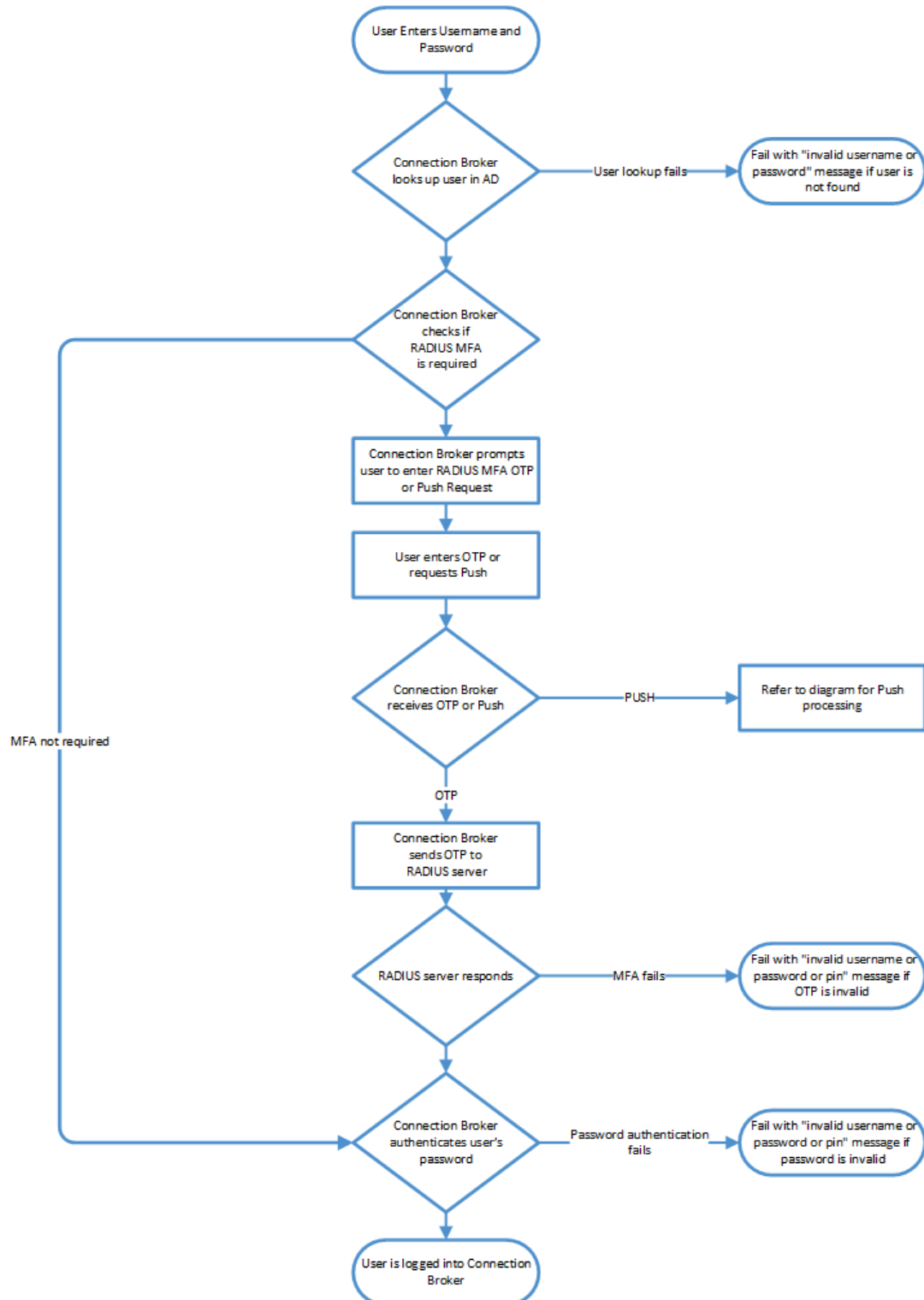
In this scenario the Connection Broker first locates the user in one of your authentication servers using the username exactly as the user typed it into the login form. This username is matched against the attribute entered in the **Match username against this field** edit field of your authentication server, typically sAMAccountName for Active Directory.

If the Connection Broker finds a matching user in your authentication server, the Connection Broker retrieves the value for the **Match username against this field** attribute as stored in your authentication server and the user's group attributes. If the user is a member of a group that requires MFA, the Connection Broker then prompts the user for the alphanumeric string to send to the RADIUS server.

The Connection Broker sends the username as returned by your authentication server and the entered alphanumeric string to the RADIUS Server. If the RADIUS server returns success *and* the authentication server password validation succeeds, the user is finally allowed to log into your Leostream environment.

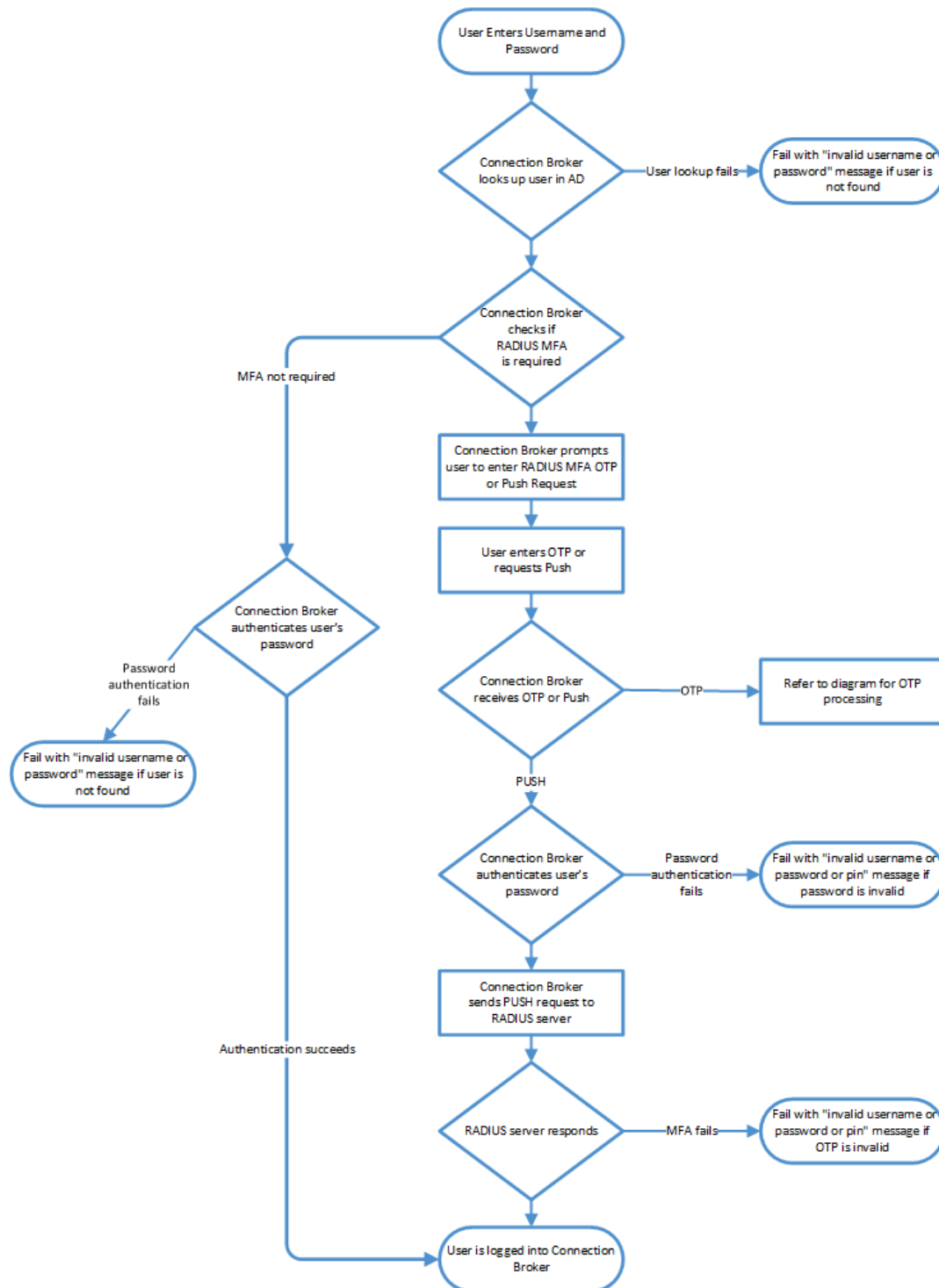
⚠ If your RADIUS server is case sensitive, ensure that the user is defined in your RADIUS server using the exact attribute value that is stored in your authentication server.

The following flow chart describes this workflow.



Requesting a Push Notification

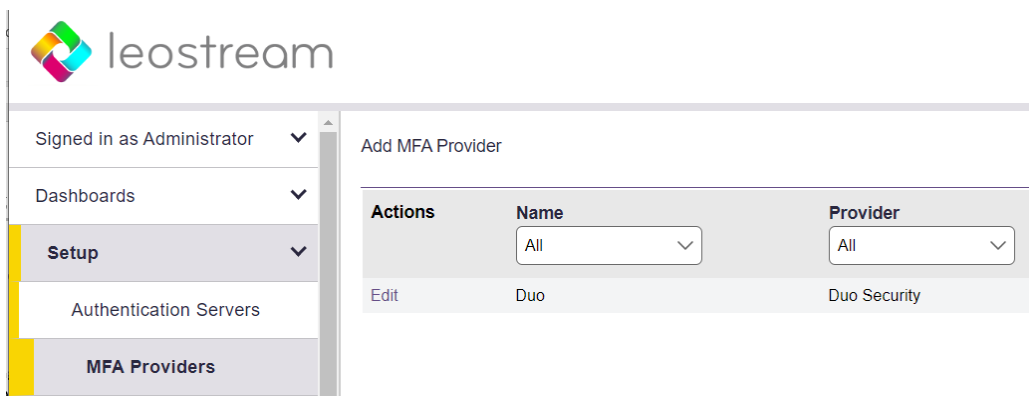
If the user requests a push notification as the second factor of authentication, the authentication work changes slightly, as described in the following diagram. In this case, the Connection Broker validates the user's password before sending the push request to the RADIUS authentication server.



Configuring the Connection Broker to Communicate with RADIUS Servers

The Connection Broker can integrate with multiple RADIUS servers, allowing you to use different identity providers for different groups of users. You integrate your Connection Broker with a RADIUS server, as follows.

1. Log into your Leostream Connection Broker Administrator web interface.
2. Go to the > **Setup** > **MFA Providers** page, shown in the following figure.



3. Click the **Add MFA Provider** link at the top of the page.
4. In the **Add MFA Provider** form, select **RADIUS Server** from the **Multi-factor Authentication Provider** drop-down menu.
5. Enter a display name for RADIUS Server in the **Name** field.
6. Enter the IP address or hostname of the RADIUS Server or Agent in the **Server IP or hostname** field.
7. In the **RADIUS port** edit field, enter the port used by your RADIUS server.



RADIUS is a UDP protocol. If your RADIUS server is behind a firewall, security group, or access control list, ensure that the RADIUS port is open for UDP traffic.

8. Specify the **RADIUS shared secret** needed to communicate with the RADIUS server or agent.
9. In the **Timeout** edit field, specify the time interval that the Connection Broker waits for the RADIUS server to reply before sending a subsequent request.



If you plan to allow users to request Push notifications, consider increasing the default timeout (30 seconds) to provide users with adequate time to receive and respond to the push notice.

10. In the **Maximum number of retries** edit field, specify the number of times the Connection Broker tries to send the RADIUS request before concluding that the RADIUS server cannot be contacted.



If you plan to allow users to request Push notifications, leave this at the default value of one retry to avoid sending the user multiple push notifications.

11. By default, the Connection Broker sends the username to the RADIUS server in the format specified by the parameter entered in the **Match login name against this field** edit field for the Active Directory or LDAP server that validates the user's password. If your usernames are formatted differently in your MFA provider, you can use the **Send username to MFA provider as** field to map the username to a different attribute from the user's Active Directory object.

For example, if user's log into Active Directory using their sAMAccountName, but the accounts in Okta are specified using email addresses, enter {EMAIL} in the **Send username to MFA provider as** field.

12. Select the **Generate Message-Authenticator attributes for Access-Requests** option if you need to use the Message-Authenticator attribute to sign packets sent to your RADIUS server. This is not common, but may be necessary if your Connection Broker returns "Failed RADIUS authentication: bad response authenticator" errors when attempting to communicate with your RADIUS server.
13. Select the **This RADIUS provider can send Push notifications** options if the identity provider associated with this RADIUS server supports push notifications by sending a passcode value of `push`. The Connection Broker automatically sends the `push` passcode to the RADIUS server when the user clicks the button to request a push notification. With this option enabled, users still have the ability to enter a one-time passcode.
14. Click **Save** on the **Add MFA Provider** form.

The Connection Broker validates the hostname and port are correct, but does not validate your shared secret when you save the form. To check if you correctly entered the shared secret, perform a test for the RADIUS server. Incorrect shared secrets result in a "bad response authenticator" error.

Testing and Troubleshooting

Use the **Test** action for the Radius server to validate that your Connection Broker can communicate with your RADIUS server.

If the test fails because your Connection Broker cannot contact the RADIUS server, double-check that your shared secret is entered correctly and that no firewall, security group, or access control list is blocking traffic from your Connection Broker to your RADIUS UDP port.

You can use the `nc` utility to scan ports and test your Connection Broker can reach your RADIUS Server, using the following command.

```
/usr/bin/nc -z -v -u RADIUS-IP 1812
```


Replace 1812 with your RADIUS server port, if you are not using the default. Note, in some cases this utility can return success even if an external firewall later blocks the traffic.

Specifying Leostream Users Who Require MFA

You use the tables on the > **Configuration > Assignments** page to control which domain users are required to pass MFA based on their AD group membership and their location. By default, no users require MFA. To enable MFA:

1. Go to the > **Configuration > Assignments** page in your Leostream Connection Broker.
2. Click the **Edit** action for the assignments table associated with the authentication server whose users require MFA.
3. Use the **MFA Provider** drop-down menu to indicate which users require MFA.

For example, in the following figure **VDI Users** who log in using Leostream Connect are not required to pass MFA in order to log into your Leostream environment. However, the same **VDI Users** logging in from a **Web Browser** do require MFA.

Edit Assignments for Authentication Server "Leodev"

Domain name
leodev.net

Assigning User Role and Policy

In this section, you can set up rules to assign Users to Roles and Policies based on their group membership. Optionally, use the Order column to re-order the rows.

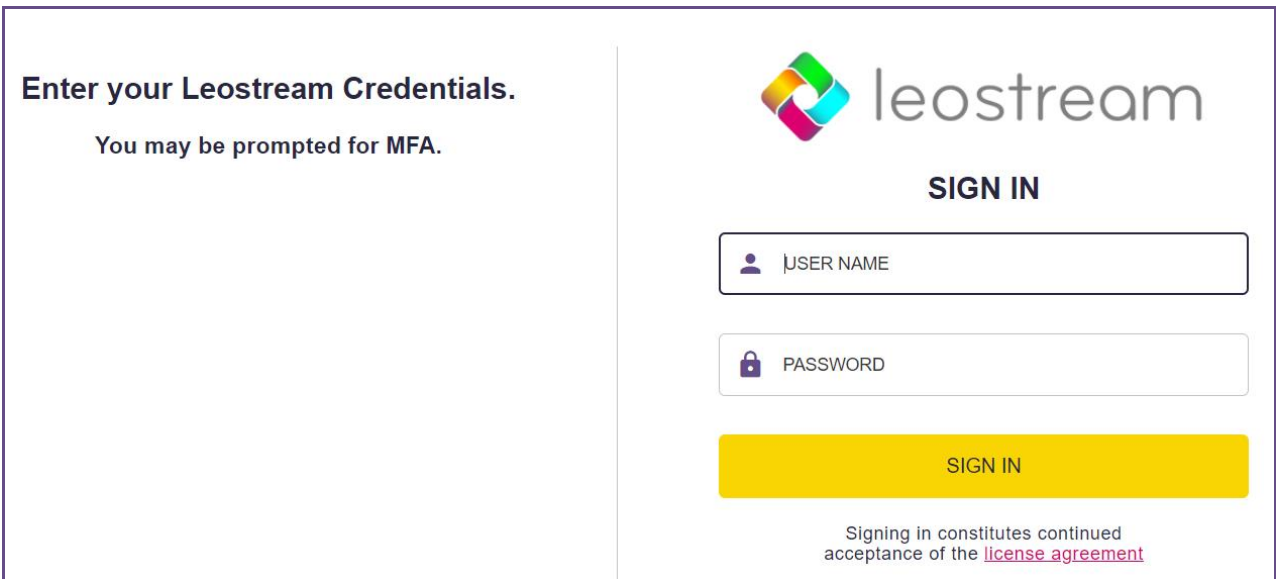
Order	Group	Client Location	MFA Provider	User Role	User Policy
1	Administrators	All	RADIUS MFA	User	Default
2	VDI Users	Web Browsers	RADIUS MFA	User	Default
3	VDI Users	Leostream Connect	<Not required>	User	Default
4		All	<Not required>	User	Default

End-User Login Workflow

Example Using the Leostream Web Client

When logging in using the Leostream Web client, users whose logins are protected by MFA must complete a second authentication step prior to receiving their offered resources. To start the Leostream login process, users first go to their Leostream Web portal, for example:


Enter your Leostream Credentials.
You may be prompted for MFA.

 **leostream**
SIGN IN

SIGN IN
Signing in constitutes continued acceptance of the [license agreement](#)

After the user enters their credentials and clicks **SIGN IN**, the Connection Broker locates the user in your authentication server, for example Active Directory. If the Connection Broker locates the user in your authentication server and determines that MFA is required, it prompts the user for MFA, for example:

Enter your Leostream Credentials.
You may be prompted for MFA.

 **leostream**
RADIUS AUTHENTICATION REQUIRED

REQUEST PUSH **SUBMIT CODE**
Signing in constitutes continued acceptance of the [license agreement](#)

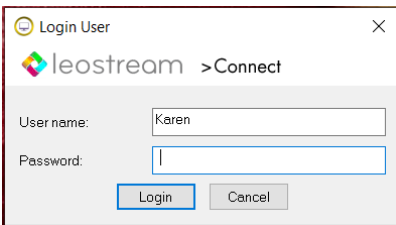
Only after the user successfully passes the MFA step will the Connection Broker display the user's offered desktops.

Example Using Leostream Connect

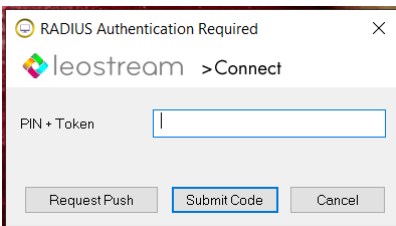
All versions of Leostream Connect support RADIUS MFA. However, older versions require users enter their username, password, and PIN/ TOTP in a single dialog and do not support Push notifications. Versions of Leostream Connect that are available with Connection Broker 9.0.40 and later support two-step authentication with Push notifications. These versions are:

- Leostream Connect for Microsoft Windows operating systems: 4.3 and later
- Leostream Connect for Linux and macOS: 3.7 and later

When logging in using Leostream Connect, users are first prompted for their username and password, as shows for Windows operating systems in the following figure.



If the user requires MFA, they are then prompted to enter a PIN + Token, for example:



Customizing the Login Dialogs

You can modify the title and prompt displayed to users that require MFA using the **Skins** in your Connection Broker, as follows.

1. Go to the > **System** > **Skins** page.
2. Click **Define Skin** to start a new terminology set.
3. Enter a descriptive name in the **Name** edit field.
4. In the **Text** tab, shown in the following figure, edit the following fields:
 - a. **MFA page title:** Defines the title of the form
 - b. **MFA verification code prompt:** Defines the text entered into the edit field
 - c. **MFA “Request push” button text:** Defines the text on the button that requests the push notice from the RADIUS server
 - d. **MFA “Submit code” button text:** Defines the text on the button that submits the OTP to the RADIUS server
5. Click **Save** to save the new terminology set.
6. Go to the > **System** > **Settings** page.
7. Select your new skin from the **Web client skin** drop-down menu in the **Web Browser Configuration** section.

Example Configuration: Okta

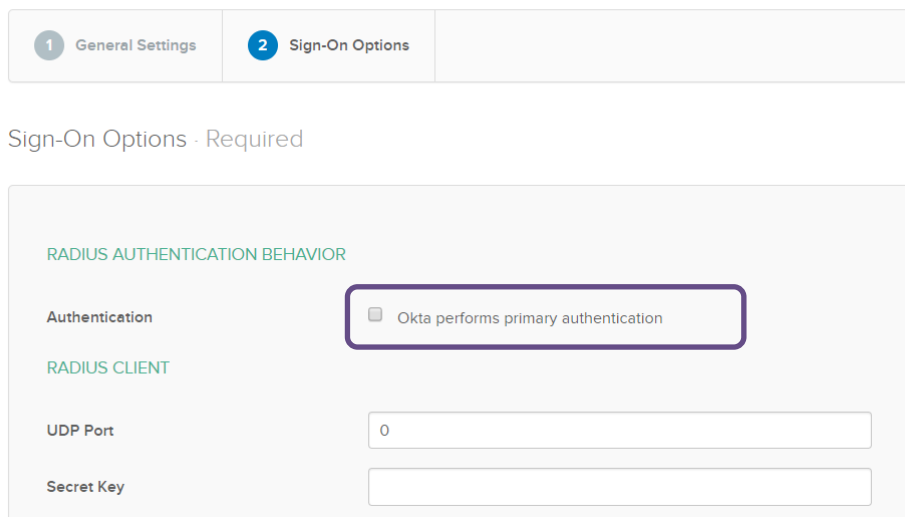
Leostream supports MFA using Okta when you deploy an Okta RADIUS Agent. For information on installing the Okta RADIUS Agent, please refer to the [Okta documentation](#). When installing the Okta RADIUS Agent, if prompted for a shared secret, make note of the secret for use when configuring Okta with your Leostream platform.

After you install the Okta RADIUS Agent, add a RADIUS Application in Okta by browsing the App Catalog. You can add a RADIUS Application in Okta by browsing the App Catalog.

For a complete description of using Okta with RADIUS integrations and how to configure a RADIUS application in Okta, please consult the [Okta documentation](#). When adding your RADIUS application in Okta, please ensure you configure the following settings appropriately.

1. Disable **Okta performs primary authentication**, as shown in the following figure.

 Add RADIUS Application



1 General Settings 2 Sign-On Options

Sign-On Options - Required

RADIUS AUTHENTICATION BEHAVIOR

Authentication ☐ Okta performs primary authentication

RADIUS CLIENT

UDP Port

Secret Key

2. In the RADIUS CLIENT section, shown in the previous figure, enter the port number and shared secret used for your Okta RADIUS Agent. If you did not specify a secret key when installing the Okta RADIUS Agent, enter a shared secret that you will use when configuring RADIUS integrations in your Leostream platform.

RADIUS is a UDP protocol. If your RADIUS Agent is behind a firewall, security group, or access control list, ensure that the RADIUS port is open for UDP traffic originating from your Connection Broker.

3. When setting the **Application username format**, ensure that the username format in Okta exactly matches the username format for your authentication server.
4. You must pre-enroll users and assign them to your Leostream application in Okta before the user can log into your Leostream environment. Self-enrollment is not currently supported.

Example Configuration: Duo

The Leostream platform has built-in integration with Duo for users logging into the Leostream Web client. See the guide for using [Duo MFA for Leostream Logins](#) for more information. For users logging in using Leostream Connect or a PCoIP Client, you can integrate with Duo using RADIUS, as described in this section.

To begin, create a RADIUS application in your Duo account.

1. From the **Applications** page, click the **Protect an Application** button.
2. Enter `Radius` into the search field to locate RADIUS in the applications list.
3. Click **Protect** for the RADIUS application.
4. Copy the **integration key**, **secret key**, and **API hostname** for your RADIUS application.

After you create the RADIUS application in Duo, install a local Duo proxy service in your environment, as described in the [Duo documentation](#). The Connection Broker uses the *RADIUS Duo only* method to perform the second authentication factor with Duo. Therefore, after installing the Duo proxy service, edit the `authproxy.cfg` file to ensure it is compatible with the Leostream authentication workflow.

First, because Leostream performs the primary authentication using Active Directory, instead of via the Duo proxy service, remove all client sections from the `authproxy.cfg` file. Then, add a `[radius_server_duo_only]` section, as described in the [Duo documentation](#).

The following is a simple `authproxy.cfg` file that shows the required information.

```
; Complete documentation about the Duo Auth Proxy can be found here:
; https://duo.com/docs/authproxy_reference

; MAIN: Include this section to specify global configuration options.
; Reference: https://duo.com/docs/authproxy_reference#main-section
[main]
debug=true

; CLIENTS: Include one or more of the following configuration sections.
; To configure more than one client configuration of the same type, append a
; number to the section name (e.g. [ad_client2])

; SERVERS: Include one or more of the following configuration sections.
; To configure more than one server configuration of the same type, append a
; number to the section name (e.g. radius_server_auto1, radius_server_auto2)

[radius_server_duo_only]
ikey=<enter integration key>
skey=<enter secret key>
api_host=<enter API hostname>
radius_ip_1=<enter Leostream hostname or IP address>
radius_secret_1=<enter shared secret for Leostream MFA provider record>
port=<enter the port to use, typically 1812>
api_timeout=60
```



If you are running a cluster of Connection Brokers, add a `radius_ip_#` and `radius_secret_#` for each Connection Broker in your cluster, using same shared secret for every IP/secret pair.

Example Configuration: Azure AD MFA and Microsoft NPS

The Leostream platform supports MFA with Azure AD Multi-Factor Authentication by leveraging a Microsoft Windows Server with the Network Policy and Access Services (NPAS) server role installed as a RADIUS server. This example shows you how to configure the Network Policy Server to integrate with your Leostream Connection Broker via the RADIUS protocol. Before you begin, please complete the following prerequisites.

- Ensure that your users are properly licensed in Office365 and have Multifactor Authentication set to **Enforced**.
- Set users' preferred MFA option to **Notify me through app**.
- Install the Network Policy and Access Services (NPAS) server role on a Windows Server machine, either using the Server Manager Dashboard to add the role or by executing the following PowerShell command on the server.

```
Install-WindowsFeature NPAS -IncludeManagementTools
```

- On the server running the NPAS role, open the **Network Policy Server console** and verify that the NPS is registered with your Active Directory.
- Verify that the Windows Firewall allows incoming RADIUS connections on the server running the NPAS role. The default RADIUS port is UDP port 1812.
- Obtain the Azure Tenant ID associated with your Azure Active Directory. You can find this ID in the **Overview** section for your Azure Active Directory service in the Azure portal.

After you complete the prerequisites, on the Windows Server with the NPAS role, configure a new RADIUS client in the Network Policy Server console, as follows.

Step 1: Create a new RADIUS client

Create a RADIUS client in NPS for your Connection Broker. If you are running a cluster of Connection Brokers, you must create a RADIUS client for each Connection Broker in the cluster.

1. Open the **Network Policy Server (NPS) console**, which can be done from the **Tools** menu of the **Server Manager**.
2. In the NPS console, expand the **RADIUS Clients and Servers** folder.
3. Right-click **RADIUS Clients** and select **New RADIUS Client**.

4. In the **New RADIUS Client** form, check the **Enable this RADIUS client** check box.
5. In the **Friendly name** edit field, enter the name to use for the RADIUS client that communications with your Leostream Connection Broker, for example `LeostreamBroker`.



If you are running a cluster of Connection Brokers, enter a series of friendly names that are easily identified using regular expressions, for example `LeostreamBroker1`, `LeostreamBroker2`, `LeostreamBroker3`, etc.

6. In the **Address** edit field, enter the IP address or FQDN of one of your Connection Brokers.
7. Ensure that **None** is selected from the **Select an existing Shared Secrets template**.
8. Select **Manual** as the method for entering a shared secret for the RADIUS client.
9. In the **Shared secret** and **Confirm shared secret** edit fields, enter a strong Shared secret that you will use when registering this RADIUS client with your Leostream Connection Broker. The **New RADIUS Client** form appears similar to the following figure.



If you are running a cluster of Connection Broker, ensure that you set the same shared secret for the RADIUS Client associated with each Connection Broker in the cluster.



Ensure that you note all **Friendly names** and the **Shared secret** for later use.

10. Click **OK**.

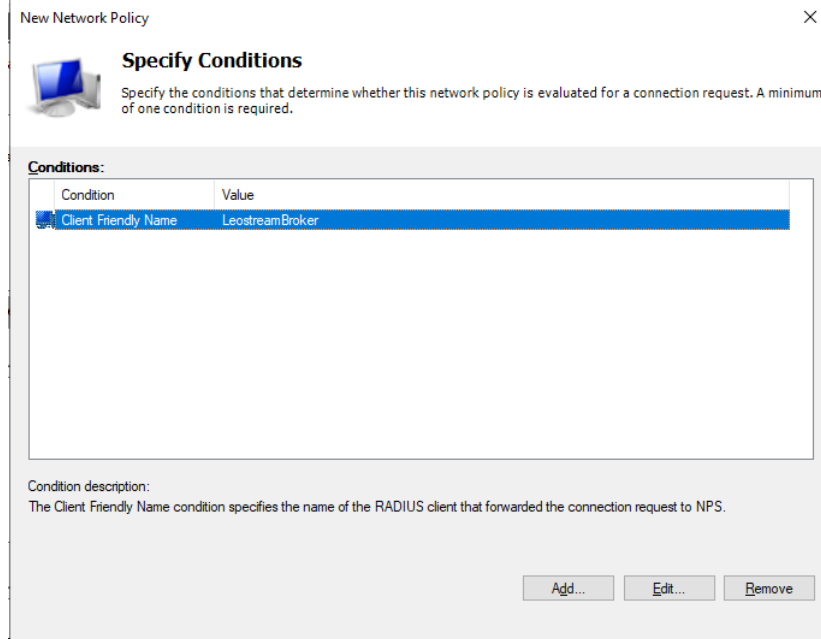
11. If you are using a cluster of Connection Brokers, repeat this process for every Connection Broker in your cluster.

Step 2: Add a new Network Policy

1. Open the **NPS console**
2. In the NPS console, expand the **Policies** node.
3. Right-click on the **Network Policies** folder and select **New**. The **New Network Policy** wizard opens.
4. In the **Policy name** edit field, enter a descriptive name for the policy, for example **Azure MFA with Leostream**, as shown in the following figure.

5. Ensure the **Type of network access** server is set to **Unspecified** and click **Next**.
6. In the **Add Conditions** page, click the **Add** button to define a new condition. The **Select condition** dialog opens.
7. In the **Select condition** dialog, select **Client Friendly Name** and click **Add**.
 - a. If you are running a standalone Connection Broker, enter the **Friendly name** you defined in Step 1 and click **OK**.
 - b. If you are running a cluster of Connection Brokers, use a regular expression for the **Friendly Name** that includes all friendly names defined in Step 1.

You now have one defined condition, as shown in the following figure.



New Network Policy

Specify Conditions

Specify the conditions that determine whether this network policy is evaluated for a connection request. A minimum of one condition is required.

Conditions:

Condition	Value
Client Friendly Name	LeostreamBroker

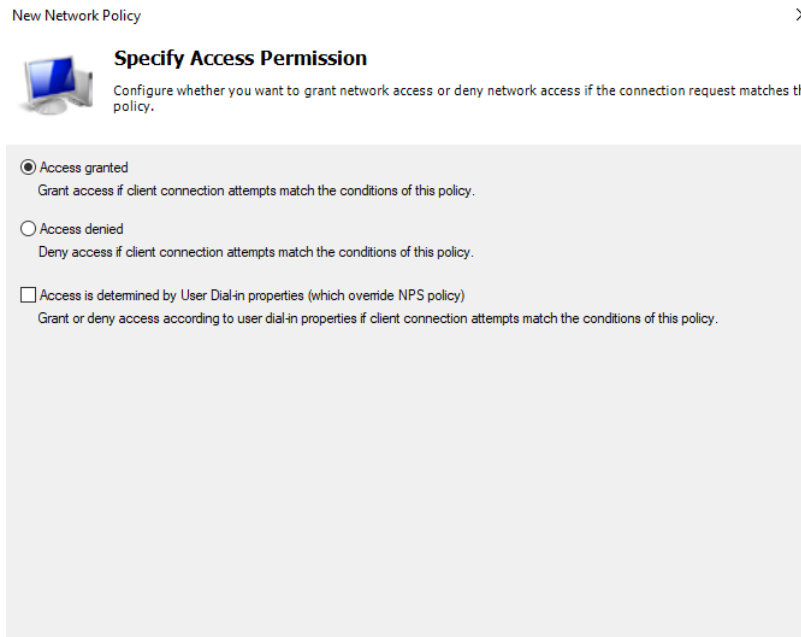
Condition description:
The Client Friendly Name condition specifies the name of the RADIUS client that forwarded the connection request to NPS.

Add... Edit... Remove



You can optionally add other conditions, for example on the user's security groups to restrict the use of this RADIUS MFA client to only certain groups of users.

8. Click **Next**.
9. On the next page, select the **Access granted** option, as shown in the following figure, and click **Next**.



New Network Policy

Specify Access Permission

Configure whether you want to grant network access or deny network access if the connection request matches the policy.

☒ **Access granted**
Grant access if client connection attempts match the conditions of this policy.

☐ **Access denied**
Deny access if client connection attempts match the conditions of this policy.

☐ **Access is determined by User Dial-in properties (which override NPS policy)**
Grant or deny access according to user dial-in properties if client connection attempts match the conditions of this policy.

10. In the **Configure Authentication Methods** page, allow all authentication methods by selecting all of the checkboxes, as shown in the following figure.

New Network Policy

Configure Authentication Methods

Configure one or more authentication methods required for the connection request to match this policy. For EAP authentication, you must configure an EAP type.

EAP types are negotiated between NPS and the client in the order in which they are listed.

EAP Types:

Move Up
Move Down

Add... Edit... Remove

Less secure authentication methods:

- ☒ Microsoft Encrypted Authentication version 2 (MS-CHAP_v2)
 - ☒ User can change password after it has expired
- ☒ Microsoft Encrypted Authentication (MS-CHAP)
 - ☒ User can change password after it has expired
- ☒ Encrypted authentication (CHAP)
- ☒ Unencrypted authentication (PAP, SPAP)
- ☒ Allow clients to connect without negotiating an authentication method.

Previous **Next** Finish Cancel

11. Click **Next**. In the information dialog that opens, shown in the following figure, click **No**.

Connection Request Policy

i You selected one or more insecure authentication methods. To ensure that each protocol is correctly configured for the remote access, policy, and domain levels, follow the step-by-step procedures in Help.

View the corresponding Help topic?

Yes **No**


12. Click **Next** on the remaining pages of the wizard until you reach the final page.

13. Click **Finish** to finalize your Network Policy.

Step 3: Add a Connection Request Policy

1. Open the **NPS console**
2. In the NPS console, expand the **Policies** node.
3. Right-click on the **Connection Request Policies** folder and select **New**. The **New Connection Request Policy** wizard opens.
4. In the **Policy name** edit field, enter a descriptive name for the policy, for example **Secondary Authentication**, as shown in the following figure.

New Connection Request Policy ×

 **Specify Connection Request Policy Name and Connection Type**
You can specify a name for your connection request policy and the type of connections to which the policy is applied

Policy name:

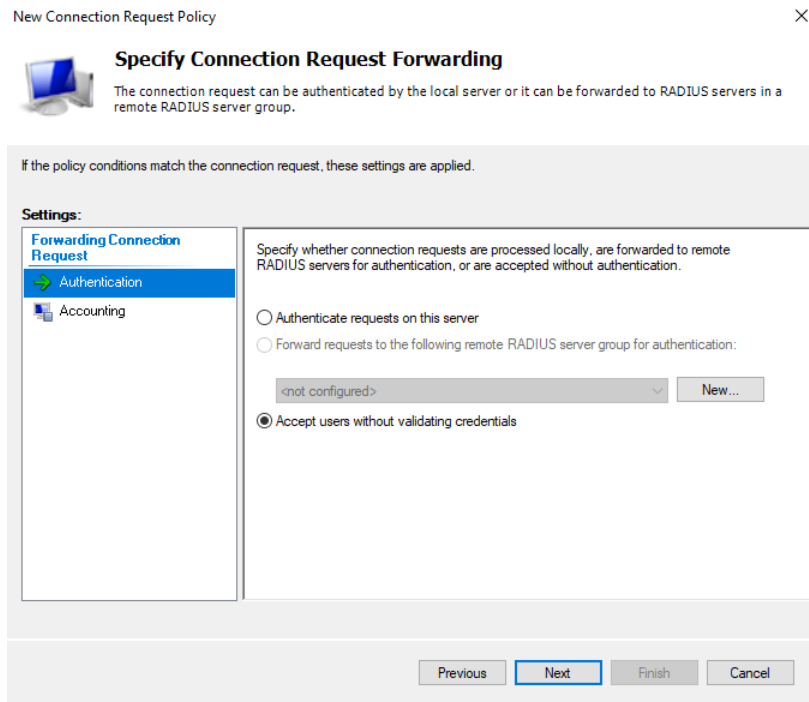
Network connection method
Select the type of network access server that sends the connection request to NPS. You can select either the network access server type or Vendor specific, but neither is required. If your network access server is an 802.1X authenticating switch or wireless access point, select Unspecified.

☒ **Type of network access server:**

☐ **Vendor specific:**

Previous **Next** Finish Cancel

5. Ensure the **Type of network access server** is set to **Unspecified** and click **Next**.
6. Define a **Client Friendly Name** condition, as describe in steps six (6) and seven (7) of the procedure for adding a new Network Policy.
7. Click **Next**.
8. Because the Connection Broker authenticates the user with their password before RADIUS MFA is performed, send only the username to trigger Azure AD MFA. To send only the username, select **Accept users without validating credentials**, as shown in the following figure.



9. Click **Next** on the remaining pages of the wizard until you reach the final page.
10. Click **Finish** to finalize your Network Policy

Step 4: Install and configure NPS Extension for Azure MFA

On the Windows Server running the NPAS role, install and configure the NPS Extension for Azure MFA.

1. Download the extension from the following Microsoft site.

<https://www.microsoft.com/en-us/download/details.aspx?id=54688>

2. Run the installer.
3. After the installation completes, run the PowerShell script from C:\Program Files\Microsoft\AzureMfa\Config (where C:\ is your installation drive), for example:

```
cd "C:\Program Files\Microsoft\AzureMfa\Config"
.\AzureMfaNpsExtnConfigSetup.ps1
```

As the script runs:

- Authenticate with your Azure Admin Account when prompted.
- Provide your Azure Active Directory Tenant ID when prompted.

4. Microsoft now enforces a number challenge with Microsoft Authenticator Push Notifications, which causes issues with the default NPS Extension settings. To override this number challenge, you'll need to add the below registry value:

```
Location: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AzureMfa
Type: STRING
Parameter: OVERRIDE_NUMBER_MATCHING_WITH_OTP
Value: FALSE
```

5. After the above changes, reboot the machine.

Step 5: Add a RADIUS MFA provider to the Connection Broker

In the Connection Broker, add a new RADIUS MFA Provider with the NPS server information, as described in [Configuring the Connection Broker to Communicate with RADIUS Servers](#), for example:

Add MFA Provider

Multi-factor Authentication provider
RADIUS Server

Name
NPS

Server IP or hostname
nps.leodev.net

Port
1812

RADIUS shared secret

Timeout (seconds)
30

Maximum number of retries
1

Send username to MFA provider as
{USER}

☐ Generate Message-Authenticator attributes for Access-Requests
☒ This RADIUS provider can send Push notifications

Notes

Save Cancel

Users assigned to this RADIUS MFA provider in your **> Configuration > Assignments** tables must then authentication using Azure AD MFA.