BeyondTrust

Privileged Remote Access
Privileged Identity Integration

 TC:2/20/2019

Table of Contents

BeyondTrust Privileged Remote Access Integration with Privileged Identity

IMPORTANT!

You must purchase this integration separately from both your BeyondTrust Privileged Remote Access and Privileged Identity solutions. For more information, contact BeyondTrust sales.

BeyondTrust's Privileged Remote Access plugin integration with Privileged Identity enables automatic password injection to authorized systems through encrypted BeyondTrust connections, removing the need to share and expose credentials to privileged accounts. In addition to the retrieval and automatic rotation of standard credentials, the integration also has the ability to retrieve shared credential lists, giving domain admins and other privileged users access to those credentials for use on the targeted systems.

Note: *Auto-rotation occurs only if configured.*

The integration between BeyondTrust PRA and PI enables:

- One-click password injection and session spawning
- Credentials never exposed to authorized users of BeyondTrust
- Access to systems on or off the network with no pre-configured VPN or other routing in place
- Passwords always stored securely in the Privileged Identity server

The BeyondTrust Endpoint Credential Manager (ECM) enables the communication between Privileged Identity and Privileged Remote Access. The ECM is deployed to a hardened Windows Server inside the firewall, typically in the same network as Privileged Identity. Once the ECM is deployed, BeyondTrust users see a list of administrator-defined credentials for the endpoints they are authorized to access. A set of these credentials can be selected when challenged with a login screen during an access session, and the user is automatically logged in, having never seen the username/password combination.

Privileged Identity handles all elements of securing and managing the passwords, so policies that require the password to be rotated after use are supported with additional configuration provided by the plugin. Privileged Remote Access handles creating and managing access to the endpoint and then recording the session and controlling the level of access granted to the user, including what the user can see and do on that endpoint.

Prerequisites for the BeyondTrust Privileged Remote Access Integration with Privileged Identity

To complete this integration, please ensure that you have the necessary software installed and configured as indicated in this guide, accounting for any network considerations. The integration is provided in the form of a plugin (ZIP archive containing the necessary DLL files and other supporting files) for use within BeyondTrust's Endpoint Credential Manager (ECM). Please ensure you have acquired the proper version of the ECM to be compliant with the version of BeyondTrust Privileged Remote Access"Configure the Privileged Identity Plugin for Integration with Privileged Remote Access" on page 8.

Applicable Versions

- Privileged Remote Access: 15.x and newer
- Privileged Identity: 5.4.0 and newer

Network Considerations

The following network communication channels must be open for the integration to work properly.

Outbound From	Inbound To	TCP Port #	Purpose
ECM Server	Privileged Remote Access Appliance	443	ECM calls to the BeyondTrust API.
ECM Server	Privileged Identity	443	ECM calls to Privileged Identity.

Configure Privileged Identity for Integration with Privileged Remote Access

The integration requires minimal setup within Privileged Identity and should work with your existing data as it stands. The two main requirements are a delegation identity that can impersonate Privileged Identity web users and the installation of the Privileged Identity SDK Web Services.

Delegation Identity

1. Under **Delegation > Web Application Identity Impersonation Mappings**, select **Create Mapping**.
2. If an identity already exists that you would like to use for the integration, select it and skip to step 3 below. Otherwise, continue with the following steps:
    a. Click **Add Identity** and select **Explicit Identity**.
    b. Enter the desired username and password, and then click **OK**.
3. Select the desired identity and click **OK**.
4. Select the identities or roles the above user should be able to impersonate, and then click **OK**.
5. Verify the new mappings, and then click **OK** to close the dialog.

> ***Note:*** *If configuring the integration to auto-spin passwords upon check-in, the above account requires the* ***All Access*** *permission. If you are not using this feature, you can skip the steps listed below.*

1. Go to **Delegation > Web Application Global Delegation Permissions**.
2. Add the **All Access** permission.
3. Select the identities or groups on the left to assign the permission to that identity or group.
4. Check the **Ignore Password Checkout** box.
5. Click **OK**.

This permission allows users to retrieve and inject credentials regardless of whether the credential is checked out to a different user in the Privileged Identity web application. It only affects the programmatic access to checked out credentials and does not allow them to check out a credential in the web application when in use by another user.

Privileged Identity SDK Web Services

Please consult the **Privileged Identity Admin Guide** for instructions on installing and enabling the SDK Web Services. In newer versions of Privileged Identity, the SDK Web Services can be enabled directly from the Privileged Identity console in the **Manage Web Appliance** section.

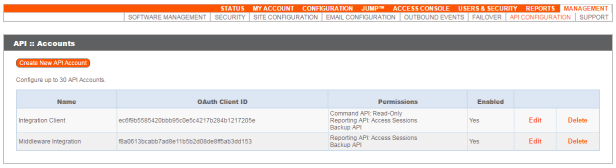
Configure Privileged Remote Access for Integration with Privileged Identity

Several configuration changes are necessary on the Privileged Remote Access Appliance to integrate with Privileged Identity.

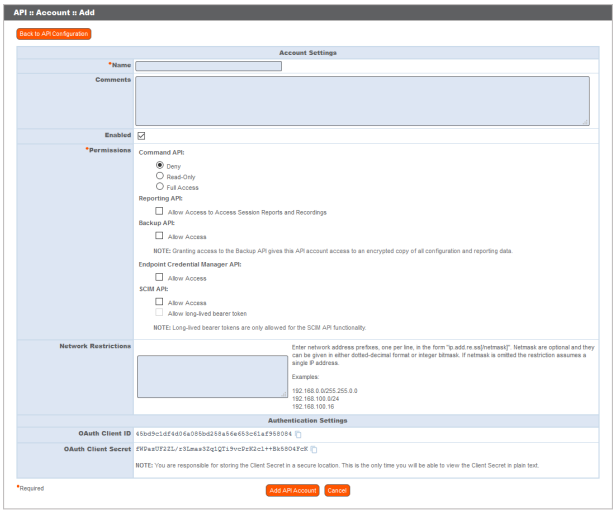
All of the steps in this section take place in the Privileged Remote Access **/login** administrative interface. Access your Privileged Remote Access interface by going to the hostname of your Privileged Remote Access Appliance followed by /login (e.g., **https://access.example.com/login**).

Create an API Service Account - Privileged Remote Access 16.2 and Later

1. Go to **Management > API Configuration** and create a new API account.



2. Under **Permissions**, check **Full Access** to the **Command API**.
3. For the **Reporting API**, check **Allow Access to Support Session Reports and Recordings** and **Allow Access to Presentation Session Reports and Recordings**. Also be sure to copy the values for both the **OAuth Client ID** and **OAuth Client Secret** for use in a later step.



4. Click **Add API Account** to create the account.

Create an API Service Account - Privileged Remote Access 16.1 and Earlier

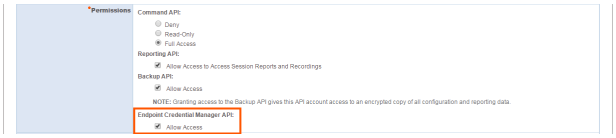
The API user account is used from within the integration to make BeyondTrust Command API calls to Privileged Remote Access.

1. Go to **/login > Users & Security > Users**.
2. Click **Create New User** and name it **Integration** or something similar.
3. Leave **Must Reset Password at Next Login** unchecked.
4. Set **Password Expires On** to **Never Expires**.
5. Scroll to the bottom and save the account.

Allow ECM Connections

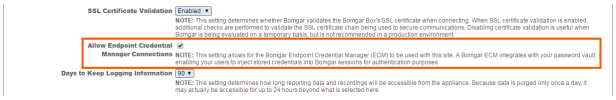
PRA 17.1 and Later

1. Go to **/login > Management > API Configuration**.
2. Add or edit an API account.
3. For **Endpoint Credential Manager API**, check **Allow Access**.



Prior to PRA 17.1

1. Go to **Management > Security**.
2. Ensure the box **Allow Endpoint Credential Manager Connections** is checked.



Configure the Privileged Identity Plugin for Integration with Privileged Remote Access

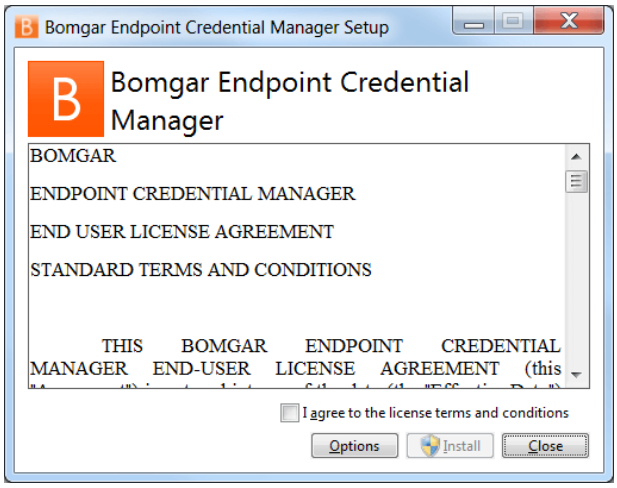
Install the Endpoint Credential Manager

The Endpoint Credential Manager (ECM) must be installed on a system with the following requirements:

- **Windows Vista or newer, 64-bit only**
- **.NET 4.5 or newer**

1. To begin, download the BeyondTrust Endpoint Credential Manager (ECM) from BeyondTrust Support at ssc.bomgar.com. Start the BeyondTrust Endpoint Credential Manager Setup Wizard.
2. Agree to the EULA terms and conditions. Mark the checkbox if you agree, and click **Install**.

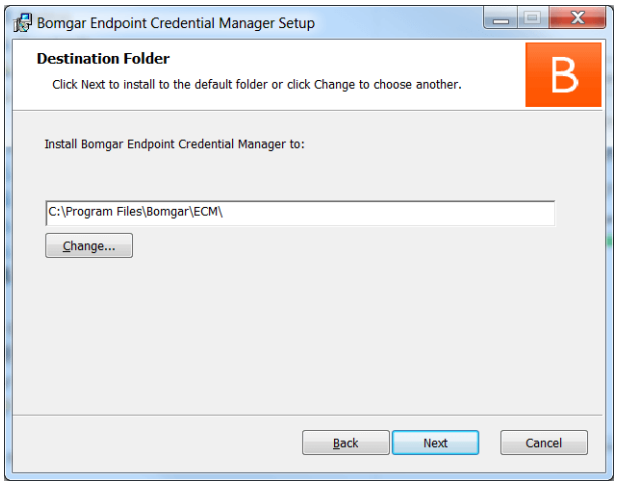
   If you need to modify the ECM installation path, click the **Options** button to customize the installation location.



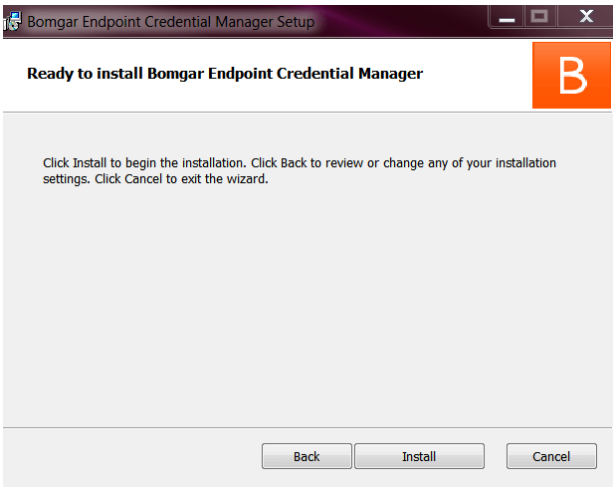
> ***Note:*** *You are not allowed to proceed with the installation unless you agree to the EULA.*

1. Click **Install**.

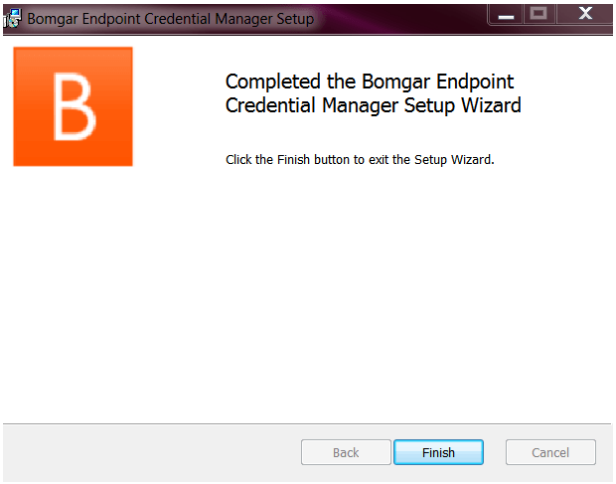
2. Choose a location for the credential manager and click **Next**.
3. On the next screen, you can begin the installation or review any previous step.



4. Click **Install** when you are ready to begin.



5. The installation takes a few moments. On the screen, click **Finish**.



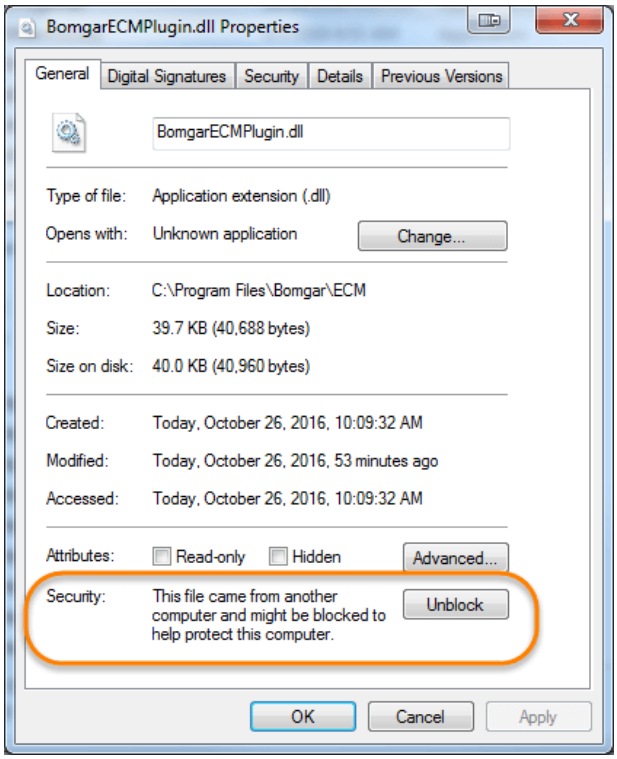
> ***Note:*** *To ensure optimal up-time, administrators can install up to five ECMs on different Windows machines to communicate with the same site on the PRA Appliance. A list of the ECMs connected to the appliance site can be found at* ***/login > Status > Information > ECM Clients****.*

> ***Note:*** *When multiple ECMs are connected to a BeyondTrust site, the PRA Appliance routes requests to the ECM that has been connected to the appliance the longest.*

Install and Configure the Plugin

1. Once the BeyondTrust ECM is installed, extract and copy the plugin files to the installation directory (typically **C:\Program Files\Bomgar \ECM**).
2. Run the **ECM Configurator** to install the plugin.

3. The Configurator should automatically detects the plugin and loads it. If so, skip to step 4 below. Otherwise, follow these steps:
    a. First, ensure that the DLL is not blocked. Right-click on the DLL and select **Properties**.
    b. On the **General** tab, look at the bottom of the pane. If there is a **Security** section with an **Unblock** button, click the button.
    c. Repeat these steps for any other DLLs packaged with the plugin.
    d. In the Configurator, click the **Choose Plugin** button and browse to the location of the plugin DLL **BomgarPIPlugin.dll**.
4. After selecting the DLL, click the gear icon in the Configurator window to configure plugin settings.

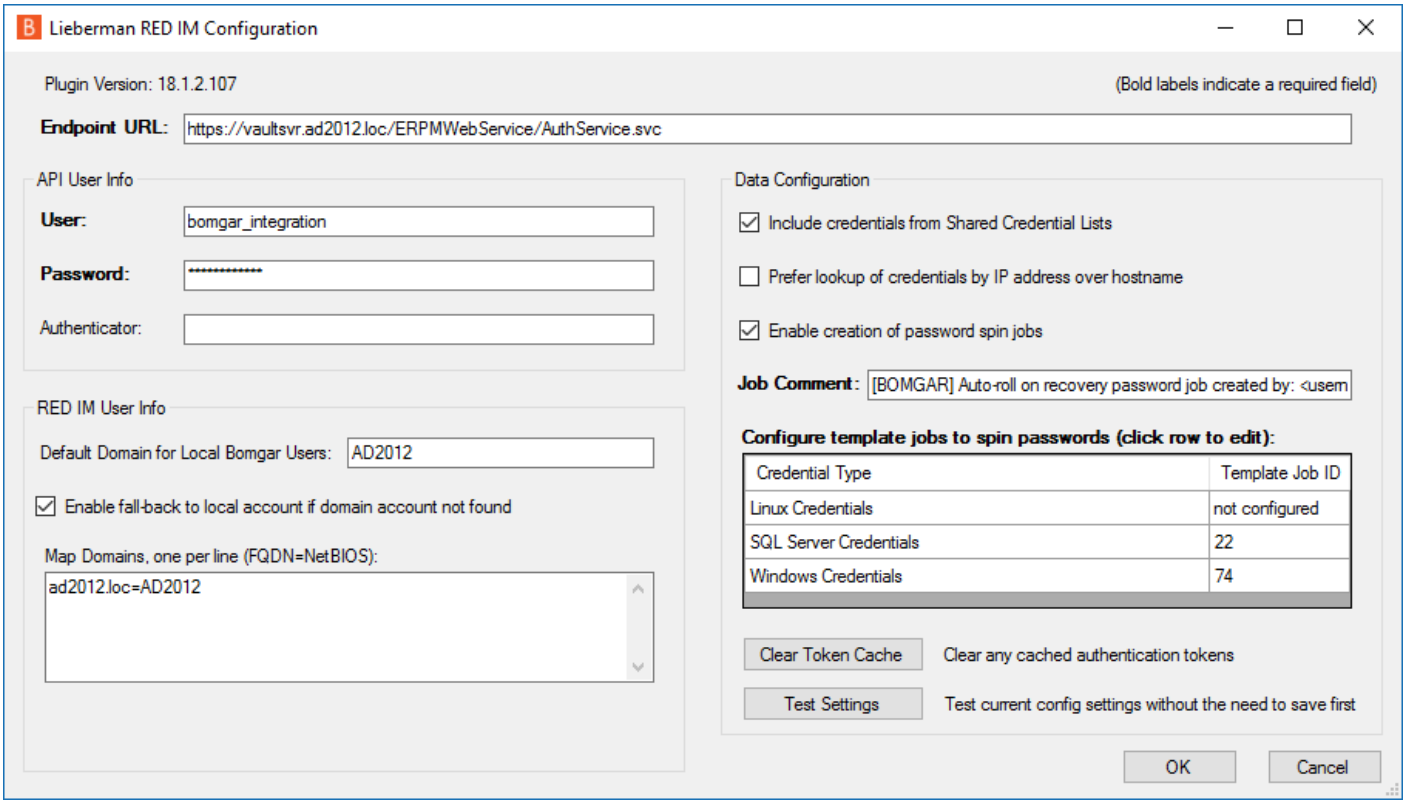


5. The following settings are available:

Setting Name	Description	Notes	Required
Endpoint URL	The full URL to the PI SDK Web Services	e.g., https://<pi-server-hostname>/ERPMWebService/AuthService.svc	Yes
API User	Delegation identity created. Assign impersonation permissions for various other PI identities and/or roles		Yes
API Password	Password of the above delegation identity		Yes
Authenticator	The authenticator associated with the delegation identity	Typically, the NETBIOS domain name for domain accounts. Leave this blank if using an explicit account.	No

Setting Name	Description	Notes	Required
Default Domain for Local BeyondTrust Users	When a value is supplied, the plugin initially attempts to retrieve credentials for the user with the username from BeyondTrust and the configured default domain	This setting is necessary if some or all PRA users are local users but the corresponding accounts in PI are domain accounts with the same username portion.	No
Enable fall-back to local account if domain account not found	When checked, the plugin first attempts to retrieve credentials for the user as a domain user and then, if no match is found, makes a second attempt without the domain	This setting is necessary if some or all BeyondTrust users are domain users but the corresponding accounts in PI are domain accounts with the same username portion.	No
Map Domains	Allows for the mapping of fully qualified domain names to their shorter NetBIOS names	This setting is necessary to match domain users in BeyondTrust to domain users in PI. BeyondTrust reports the logged-in user with the fully qualified domain name (FQDN), while PI may expect the NetBIOS name of the domain. It is also used for returning domain credentials for Windows endpoints when the domain of the endpoint is not known. These mappings must be done manually and can be entered one per line as FQDN=NetBIOS (e.g., Example.local=EX).	No
Include credentials from Shared Credential Lists	When checked, the plugin includes credentials from a shared credential list	In addition to retrieval of normal managed credentials, the integration can also retrieve endpoint-specific credentials from a shared list.	No
Prefer lookup of credentials by IP address over hostname	When checked, the plugin attempts to find credentials for the endpoint using its IP address, if available	If the IP address is not available, the plugin attempts to find credentials by using the hostname, which is the default behavior.	No
Enable creation of password spin jobs	When checked, the plugin creates password spin jobs for credentials checked out via the integration	Checking out credentials via the PI SDK Web Services does NOT result in a spin job for managed passwords that would normally rotate when checked in via the web interface. To compensate for this, the plugin can examine the credential to see if it is set to auto-spin and then create a job to do so. No spin job is created for credentials that do not have random passwords or that are not configured to auto-spin.	No

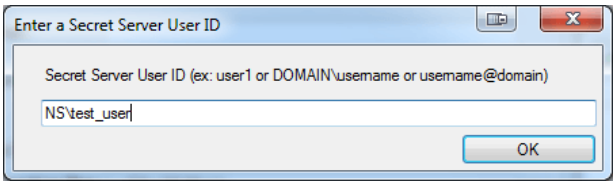
Setting Name	Description	Notes	Required
Job Comment	A custom job comment can be configured to help distinguish jobs submitted as part of the integration	The string **<username>** replaces the username with the PI identity performing the check-out. It can be replaced anywhere in the string or removed, if desired.	No
Password Change Template Job IDs	The numeric IDs of the template job shown in the Jobs list in PI	It is recommended to create password change jobs that can be used as templates for future jobs submitted by the integration. The basic settings of these jobs are used for each subsequent job with only the password, endpoint-specific information, and scheduling being overridden. There must be a separate template job created and configured for each type of stored credential you would like to rotate. ***Note:** Make sure you do not delete the template jobs.*	No



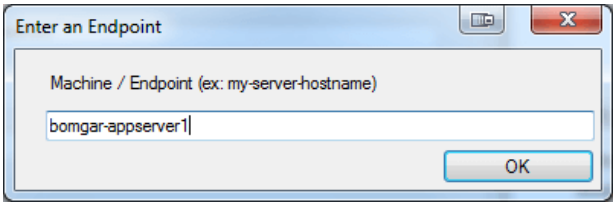
Test Settings

The settings specific to Privileged Identity can be tested directly from the plugin configuration screen using the **Test Settings** button.

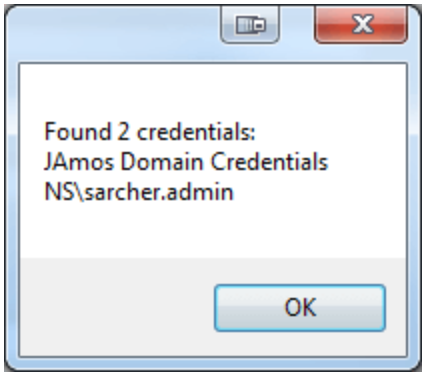
1. Enter a user account from which to retrieve credentials.



2. Enter an endpoint for which the user account has one or more credentials.



3. View the resulting list.



Note: *No actual passwords are retrieved or displayed, only the list of credentials.*

Note: *The settings used for the test are the ones currently entered on the screen, not necessarily what is saved.*

Clear Token Cache

To avoid excessive authentication calls to Privileged Identity, the plugin caches authentication tokens (in an encrypted form) for users as they attempt to retrieve secrets through the integration. Subsequent calls use the cached token until it expires. At that point, a new authentication token is retrieved and cached. The **Clear Token Cache** button allows an admin to clear all cached authentication tokens if such action becomes necessary for maintenance, testing, etc.

Troubleshoot the Privileged Remote Access and Privileged Identity Integration

To assist you, a list of common issues experienced during the integration process has been provided, and steps for resolving these issues are noted.

For any issues that involve the ECM service, it is recommended to enable **DEBUG level logging**. To enable this setting, follow these steps.

1. Open the **Bomgar-ECMService.exe** config file in a text editor.
2. Edit the file by changing the line **<level value="INFO"/>** to **<level value="DEBUG"/>**.
3. Save the file and restart the ECM service.

Common Issues and Resolution Steps

Issue	Cause	Debugging Steps/ Possible Solutions
ECM Configurator cannot find or load the plugin	DLL files were not deployed to ECM install directory.	Copy **ALL** files included with the plugin into the ECM install directory, typically C:\Program Files\Bomgar \ECM. Close and re-open the ECM Configurator.
ECM Configurator cannot find or load the plugin	DLL files are being blocked by Windows.	While the build server signs the assemblies to help prevent this error, some systems still block the DLLs. To unblock them, right-click on the DLL. Select **Properties**. In the **General > Security** section, check the **Unblock** box. Click **OK** to save the changes. Repeat these steps with any other DLLs being paged with the plugin DLL.
No credentials are returned when using the **Test Settings** feature	ECM has been configured without the proper settings.	A failure to retrieve credentials using the **Test Settings** feature in the ECM Configurator is usually a result of some configuration setting being entered incorrectly. First, double-check any usernames and passwords entered. Next, check the logs in **Configurator.log** to see if the integration is providing any information as to why the test failed. It could be anything from incorrect URLs / ports, authentication failure, or network connectivity issues. The logs may also reveal a perceived failure was not a failure after all. Instead, no matches may have been found, and even if this is unexpected, an empty list is still a valid result. ***Note:*** *The* ***Test Settings*** *feature does* ***NOT*** *communicate with BeyondTrust PRA at any point. It simply tests the settings related to the password vault system. Also, remember that the test uses the currently entered values and settings whether the settings have been saved or not. This allows you to test different configurations without overwriting existing settings.*

Issue	Cause	Debugging Steps/ Possible Solutions
No credentials are returned when using the **Test Settings** feature	There is a lack of network connectivity.	There is a lack of necessary network connectivity between the ECM server and the password vault system. The resolution could be as simple as adding a rule to the Windows Firewall, or it may require a network administrator to open ports to allow communication.
Credentials are returned via the **Test Settings** feature but are not available in the access console	ECM has been configured without the proper settings.	The settings on the initial screen of the ECM Configurator tell the ECM service which BeyondTrust PRA instance to connect to and the account to use for authentication. Double-check these and review the logs in **ECM.log**, if necessary.
Credentials are returned via the **Test Settings** feature but are not available in the access console	BeyondTrust PRA has been configured without the proper settings.	It is possible ECM connections have not been enabled or the API account being used is not configured to be an administrator. Review the steps in "Configure Privileged Remote Access for Integration with Privileged Identity" on page 6
Credentials are returned via the **Test Settings** feature but are not available in the access console	The ECM service has stopped functioning.	Restart the BeyondTrust ECM Service.
Credentials are returned via the **Test Settings** feature but are not available in the access console	There is a lack of network connectivity.	A lack of connectivity could be preventing the integration from working. In this case, the missing connection would occur between BeyondTrust PRA and the ECM server. If the ECM is unable to establish a connection to the BeyondTrust PRA Appliance, it is unable to receive requests for credentials. Try loading the **/login** page in a browser running on the ECM server. If the browser cannot connect, the ECM will also be unable to connect. If the browser test passes, check the **ECM.log** to see if a connection was successfully established when starting the service.
Credentials are returned via the **Test Settings** feature but are not available in the access console	The user mapping has failed.	This issue commonly occurs (particularly with domain accounts) when a test is run with a user entered as domain\user or a similar format. However, when connecting through the access console, it is possible for the domain portion to be different or missing altogether. If the PRA user is a local user, no domain information is present. The same is true for users authenticating to PRA via certain security providers like RADIUS. If the plugin allows for domain mapping or default domains for local users, verify these are configured correctly. Also, check the **ECM.log** to make sure the values passed to the password vault match what is expected. If the test is successful, note the information used.

