



BeyondTrust

Privileged Remote Access Privileged Identity Integration

Table of Contents

BeyondTrust Privileged Remote Access Integration with Privileged Identity	3
Prerequisites for the BeyondTrust Privileged Remote Access Integration with Privileged Identity	4
Applicable Versions	4
Network Considerations	4
Configure Privileged Identity for Integration with Privileged Remote Access	5
Delegation Identity	5
Privileged Identity SDK Web Services	5
Configure Privileged Remote Access for Integration with Privileged Identity	6
Create an OAuth API Account	6
Allow ECM Connections	7
Configure the Privileged Identity Plugin for Integration with Privileged Remote Access	8
Install the Endpoint Credential Manager	8
Install and Configure the Plugin	10
Test Plugin Settings	15
Clear Token Cache	18
Troubleshoot the Privileged Remote Access and Privileged Identity Integration	19
Common Issues and Resolution Steps	19

BeyondTrust Privileged Remote Access Integration with Privileged Identity

IMPORTANT!

You must purchase this integration separately from your BeyondTrust Privileged Remote Access solution. For more information, contact BeyondTrust's Sales team.

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	BeyondTrust Appliance B Series	443	ECM calls to the BeyondTrust API.
ECM Server	Privileged Identity	443	ECM calls to Privileged Identity SDK Web Services.

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 three below. Otherwise, continue with the following steps:
 - Click **Add Identity**, and then select **Explicit Identity**.
 - Enter the desired username and password, and then click **OK**.
3. Select the desired identity, and then 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.

6. Go to **Delegation > Web Application Global Delegation Permissions**.
7. Add the **All Access** permission.
8. Select the identities or groups on the left to assign the permission to that identity or group.
9. Check the **Ignore Password Checkout** box.
10. 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 B Series Appliance to integrate with Privileged Identity.

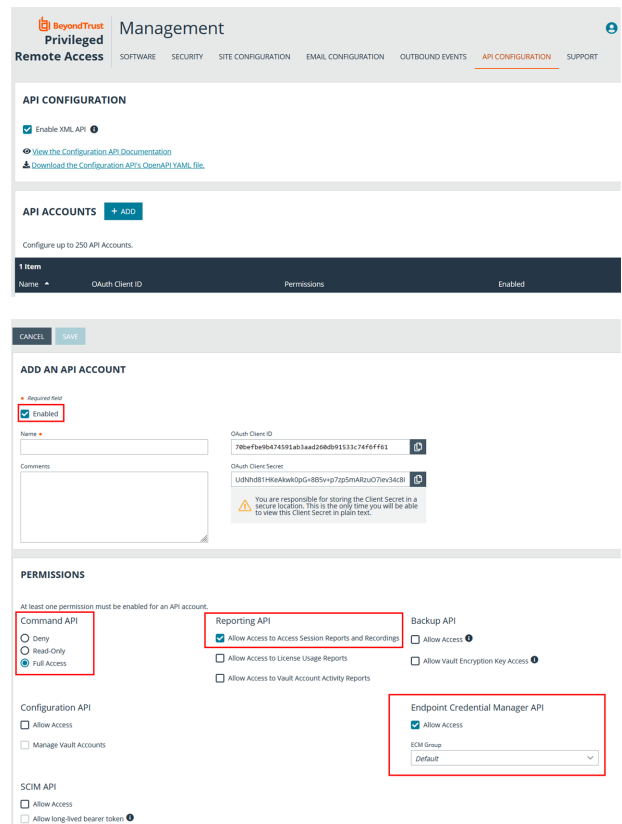
All of the steps in this section take place in the Privileged Remote Access **/login** administrative interface. Access your Privileged Remote Access interface by going to the hostname of your B Series Appliance followed by **/login** (e.g., <https://access.example.com/login>).

Create an OAuth API Account


The Privileged Identity API account is used from within Privileged Identity to make Privileged Remote Access Command API calls to Privileged Remote Access.

1. In **/login**, navigate to **Management > API Configuration**.
2. Click **Add**.

3. Check **Enabled**.
4. Enter a name for the account.
5. **OAuth Client ID** and **OAuth Client Secret** are used during the OAuth configuration step in Privileged Identity.
6. Set the following **Permissions**:
 - **Command API**: Full Access.
 - **Reporting API**: Allow Access to Access Session Reports and Recordings.
 - **Endpoint Credential Manager API**: Allow Access.
 - If ECM groups are enabled on the site, select which **ECM Group** to use. ECMs that are not associated with a group come under **Default**.



The screenshot shows the 'Management' interface for 'BeyondTrust Privileged Remote Access'. The 'API CONFIGURATION' section is active, showing 'Enable XMLE API' checked. Below, the 'API ACCOUNTS' section has an '+ ADD' button. A table shows one API account with columns for Name, OAuth Client ID, Permissions, and Enabled. The 'ADD AN API ACCOUNT' form is open, with the 'Enabled' checkbox checked. The 'Name' field is empty. The 'OAuth Client ID' and 'OAuth Client Secret' fields contain long alphanumeric strings. A warning message states: 'You are responsible for storing the Client Secret in a secure location. This is the only time you will be able to view this Client Secret in plain text.' The 'PERMISSIONS' section has a note: 'At least one permission must be enabled for an API account.' Under 'Command API', 'Full Access' is selected. Under 'Reporting API', 'Allow Access to Access Session Reports and Recordings' is checked. Under 'Backup API', 'Allow Access' is checked. Under 'Endpoint Credential Manager API', 'Allow Access' is checked and the 'ECM Group' is set to 'Default'. Other permissions like 'Configuration API', 'SCIM API', and 'Allow long-lived bearer token' are unchecked.

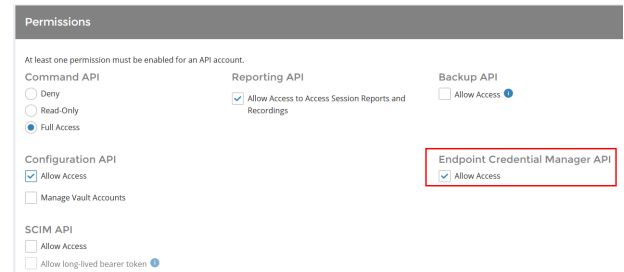
 **Note:** The ECM Group feature is only present if enabled when your site is built. If it is not present, please contact your site administrator.

7. Click **Save** at the top of the page to create the account.

Allow ECM Connections

PRA 20.1 and later

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



Permissions

At least one permission must be enabled for an API account.

Command API	Reporting API	Backup API
<input type="radio"/> Deny	<input checked="" type="checkbox"/> Allow Access to Access Session Reports and Recordings	<input type="checkbox"/> Allow Access ⓘ
<input type="radio"/> Read-Only		
<input checked="" type="radio"/> Full Access		

Configuration API	Endpoint Credential Manager API
<input checked="" type="checkbox"/> Allow Access	<input checked="" type="checkbox"/> Allow Access
<input type="checkbox"/> Manage Vault Accounts	

SCIM API
<input type="checkbox"/> Allow Access
<input type="checkbox"/> Allow long-lived bearer token ⓘ

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:

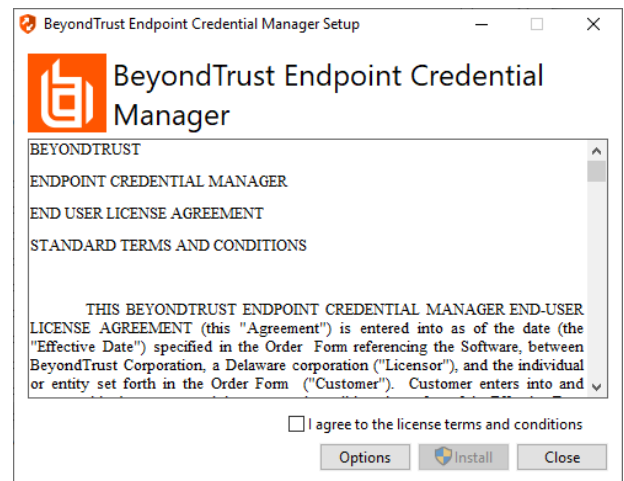
- Windows Vista or newer, 64-bit only
- .NET 4.5 or newer
- Processor: 2GHz or faster
- Memory: 2GB or greater
- Available Disk Space: 80GB or greater

1. To begin, download the BeyondTrust Endpoint Credential Manager (ECM) from [BeyondTrust Support](https://beyondtrustcorp.service-now.com/csm) at beyondtrustcorp.service-now.com/csm.
2. Start the BeyondTrust Endpoint Credential Manager Setup Wizard.
3. Agree to the EULA terms and conditions. Check the box if you agree, and then click **Install**.

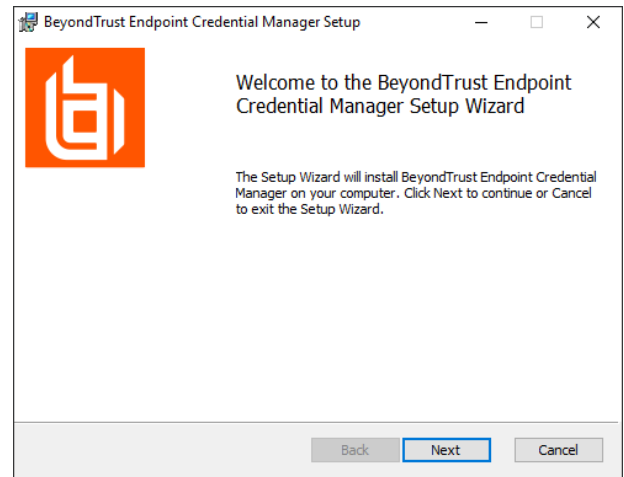
If you need to modify the ECM installation path, click the **Options** button to customize the installation location.



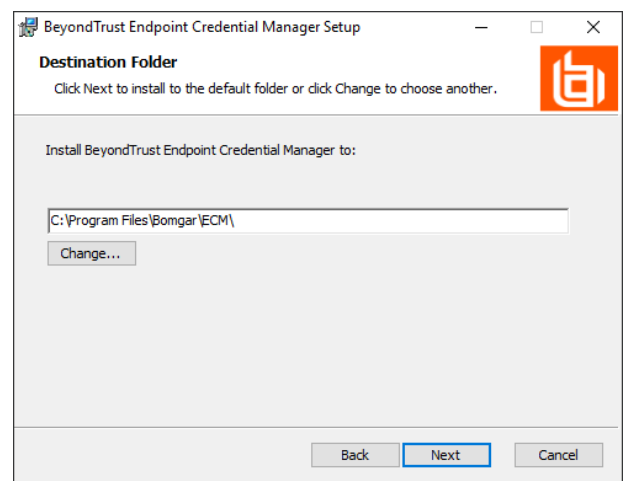
Note: You are not allowed to proceed with the installation unless you agree to the EULA.



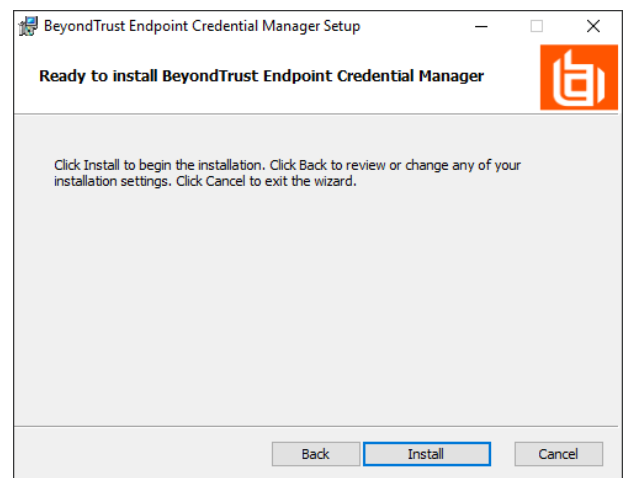
4. Click **Next** on the Welcome screen.



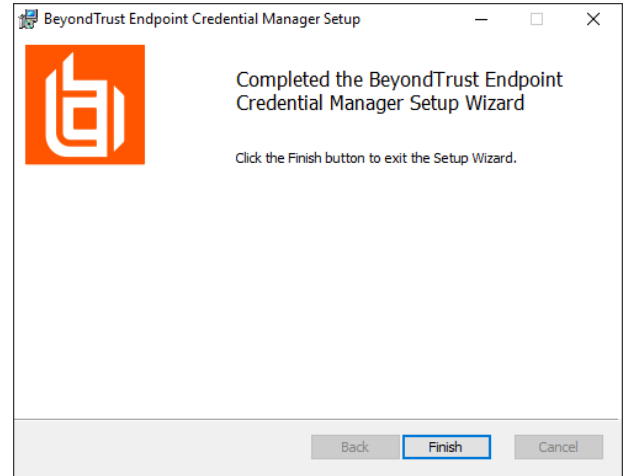
5. Choose a location for the credential manager, and then click **Next**.
6. On the next screen, you can begin the installation or review any previous step.



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



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



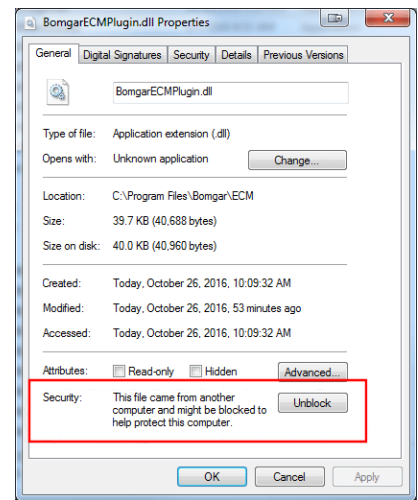
Note: To ensure optimal up-time, administrators can install up to three ECMs on different Windows machines to communicate with the same credential store. A list of the ECMs connected to the appliance site can be found at **/login > Status > Information > ECM Clients**.



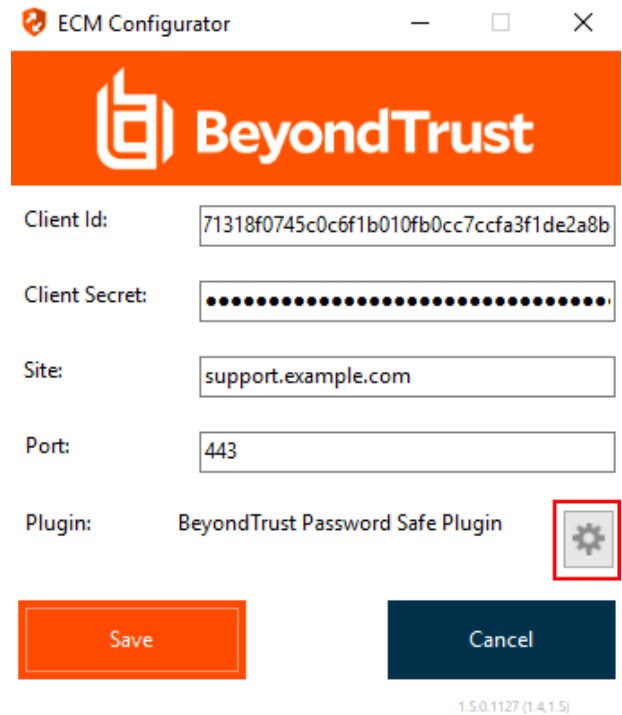
Note: When ECMs are connected in a high availability configuration, the BeyondTrust Appliance B Series routes requests to the ECM in the ECM Group that has been connected to the appliance the longest.

Install and Configure the Plugin

- Once the BeyondTrust ECM is installed, extract and copy the plugin files to the installation directory (typically **C:\Program Files\Bomgar\ECM**).
- Run the **ECM Configurator** to install the plugin.
- The Configurator should automatically detect the plugin and load it. If so, skip to step 4 below. Otherwise, follow these steps:
 - First, ensure that the DLL is not blocked. Right-click on the DLL and select **Properties**.
 - On the **General** tab, look at the bottom of the pane. If there is a **Security** section with an **Unblock** button, click the button.
 - Repeat these steps for any other DLLs packaged with the plugin.
 - In the Configurator, click the **Choose Plugin** button and browse to the location of the plugin DLL.



4. Click the gear icon in the **Configurator** window to configure plugin settings.



ECM Configurator


BeyondTrust

Client Id:

Client Secret:


Site:

Port:

Plugin: BeyondTrust Password Safe Plugin 

1.5.0.1127 (1.4.1.5)

5. The following settings are available:

 BeyondTrust Privileged Identity Configuration
✕

Plugin Version: 19.3.1.187 (Bold labels indicate a required field)

Endpoint URL:

API User Info

User:

Password:

Authenticator:

Data Configuration

Include credentials from Shared Credential Lists

Include credentials where user only has Allow Remote Session permission

If host domain is known, filter domain credentials for Windows Jump items

If host domain is known, filter local credentials for Windows Jump items

Include domain credentials for Shell Jump items

Include domain credentials for Web Jump items

Prefer lookup of credentials by IP address over hostname

Enable creation of password spin jobs Privileged Identity 5.5.3.1 or new ▾

Job Comment:

Configure template jobs to spin passwords (click row to edit):

Credential Type	Template Job ID
Linux Credentials	not configured
SQL Server Credentials	not configured
Windows Credentials	not configured

BeyondTrust PI User Info

Default Domain for Local SRA Users:

Enable fall-back to local account if domain account not found


Map Domains, one per line (FQDN=NetBIOS):

Test current config settings without the need to save first

Setting Name	Description	Notes	Required
Endpoint URL	The full URL to the PI Web Service	e.g., https://<pi-server-hostname>/ERPWebService/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
API Registration Key	The Key for the API Registration created for the integration		Yes

Setting Name	Description	Notes	Required
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
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
Global Approver	The username for the account created to allow automated approval of requests for credentials via the integration		Yes
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

Setting Name	Description	Notes	Required
Include credentials where user only has Allow Remote Session permission	When checked, the plugin includes credentials for which the user only has Remote Session permissions	This setting allows for configuration of Privileged Identity users and credentials, where certain credentials are only available to the user for injection. Passwords for these credentials cannot be checked out via the web interface. It is also important to note that, because the individual users won't have the Recover Password permission, checkout and recovery of the credential also cannot take place via the API through the integration. As a result, if the user selects one of these credentials for injection, checkout must be performed as the configured API user account. If this setting is enabled, it is critical that the API user account have sufficient permissions to perform checkouts of the desired credentials.	No
If host domain is known, filter domain credentials for Windows Jump items	When checked, the plugin will filter the domain credentials based on the domain of the Windows endpoint to which the user is connecting	This setting is enabled by default to avoid returning domain credentials for other domains that likely can't be used on the endpoint. However, there may be cases where this behavior is not desired, such as in an organization where credentials exist across multiple domains, but there is also a trust relationship established between those domains. If no domain is known or reported by the configured Jump item, the default behavior to return domain credentials from all known domains remains the same whether this setting is enabled or not.	No
If host domain is known, filter local credentials for Windows Jump items	When checked, the plugin will filter the local credentials based on the domain of the Windows endpoint to which the user is connecting	This setting allows an admin to limit the potential results returned for a give Jump item by ensuring local credential results match the FQDN of the endpoint rather than just the hostname portion (normal behavior). This additional level of filtering only takes place for Windows endpoints (RDP and Jump Client items) and when the hostname and domain are both known by the ECM and plugin.	No
Include domain credentials for Shell Jump items	When checked, the plugin will include any available domain credentials when a user initiates a connection to a Shell Jump item	This allows user to inject their domain credentials (or others available to them) into Unix/Linux based systems that support them. Credentials are provided in each of the following three formats, allowing the user to select the appropriately formatted credential for injection for each Jump item: - username (only) - username@DOMAIN - DOMAIN\username	No
Include domain credentials for Web Jump items	When checked, the plugin will include any available domain credentials when a user initiates a connection to a Web Jump item	This allows users to inject their domain credentials (or others available to them) into many sites, internal or otherwise, that support them. Credentials are provided in each of the following three formats, allowing the user to select the appropriately formatted credential for injection for each Jump item: - username (only) - username@DOMAIN - DOMAIN\username	No

Setting Name	Description	Notes	Required
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 Web Service 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
Select your PI version	Selection determines which API calls will be made for the creation of spin jobs when session ends and a credential is released	As of PI 5.5.3.1, a new API was added which streamlines the auto-spin behavior by eliminating the need to create and configure template jobs and for the integration to manually submit new spin jobs based on those templates. The integration can now provide the details of the credential to spin and PI will handle the rest internally, mirroring the same behavior that occurs when a user checks in a Store Credential via the PI web interface. Admins can select their PI version in the plugin configuration. This will dictate which API calls are made and maintains backward compatibility for PI releases prior to 5.5.3.1.	No
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	<p>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.</p> <p>There must be a separate template job created and configured for each type of stored credential you would like to rotate.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  Note: Make sure you do not delete the template jobs. </div>	No


Test Plugin Settings

You can test the settings specific to Privileged Identity directly from the plugin configuration screen using the **Test Settings** button.

The test functionality allows you to test new or updated configuration without the need to go through the access console or to save the changes first. The form collects information to simulate a request from the B Series Appliance to the ECM. This means you can test the settings without having the ECM service running or connected to the B Series Appliance.



Note: While the test does simulate a request from the B Series Appliance to the ECM, it does not in any way test configuration or connectivity to the B Series Appliance. It is used only for configuration, connectivity, permissions, etc., related to the password vault system.

 Test Plugin Settings
✕

This form provides a way to test new or updated configuration without the need to first save the changes. Also, because the test simulates a request from the Secure Remote Access appliance, it doesn't require the ECM service to be connected to the appliance or even running at all.

(Bold labels indicate a required field)

Console User Information

Simulates the console user information that would be sent to the ECM from the appliance

SRA Console Username:

Distinguished Name:

Jump Item Information

Simulates the Jump Item to which a user would connect and attempt credential injection

Jump Item Type:

Hostname / IP Address:

Additional IP Address:

Application Name:

NOTE: Any logs generated from these tests will be contained in Configurator.log

Console User Information

The fields collected in this section simulate the information that is sent to the ECM about the user logged into the console and requesting credentials from the password vault.

- **SRA Console Username:** The username of the console user. Depending on the type of security provider and how it is configured, this might be username-only (**joe.user**), which is the most common format, or it might include other information and in other formats, such as down-level domain info (**ACMEjoe.user**) or email / UPN (**joe.user@acme-inc.com**).
- **Distinguished Name:** For LDAP Security Providers, the provider often populates the Distinguished Name of the user in addition to the username. The Distinguished Name includes domain information which is extracted by the integration and used to help identify the matching account in the password vault. An example DN is: **uid=joe.user,ou=HelpDesk,dc=acme-inc,dc=com**.

Jump Item Information

The fields collected in this section simulate the information that is sent to the ECM about the endpoint or Jump Item to which the console user may connect.

- **Jump Item Type:** Because different Jump Items result in different pieces of information being sent to the ECM, as well as how the ECM may query the password vault for applicable credentials, it is important to identify the type of Jump Item you wish to simulate

as part of the test process.



Note: The Jump Client type should be used to simulate Remote Jump and Local Jump items as well.

- **Hostname / IP Address:** For most types of Jump Items, the primary piece of information used to find credentials in the password vault is the endpoint's hostname or IP address.
- **Website URL:** For Web Jump items, rather than a hostname, the ECM is provided with the URL to which the item points. This field validates that the supplied string appears to be an actual URL.
- **Additional IP Address:** For Jump Client items, in addition to the machine's name, the installed client also makes the machine's public and private IP addresses available to the ECM. Some integrations use this information to query for credentials in addition to or even instead of those which match the hostname value.
- **Application Name:** For testing credential retrieval for injection into an application via an RDP + SecureApp item, the ECM is provided with both a value to identify the endpoint (Hostname / IP Address) and one to identify the specific application. The required value for Application Name may vary across integrations. The integration specific installation guides should contain more information on possible values.

Test Results

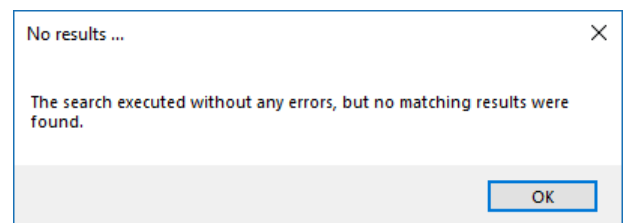
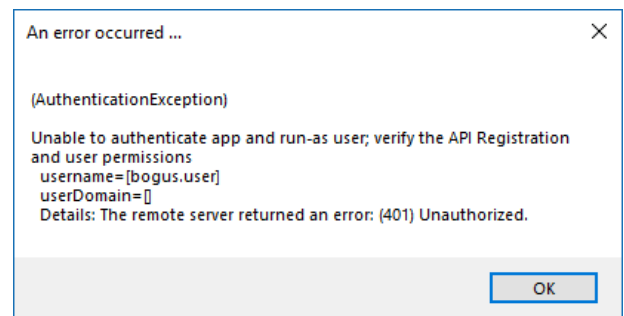
If the test fails for any reason, error information is displayed to assist in diagnosing the cause of the failure. In most cases these errors are handled and then assigned a type, such as an authentication-related error, and then displayed with the inputs as well as any specific error messages. However, there may still be some instances where a particular error might not be anticipated, so the information is displayed in a more raw form.



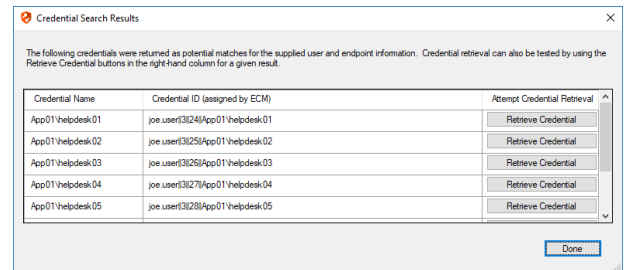
Note: It's important to note that, either way, the same information is included in the **Configurator.log**, along with more detail as to exactly what point in the execution the failure occurred.

It's possible that the test succeeds in that it doesn't encounter any errors and yet it doesn't return any credentials. Because this is a perfectly valid result, it is not treated as an error.

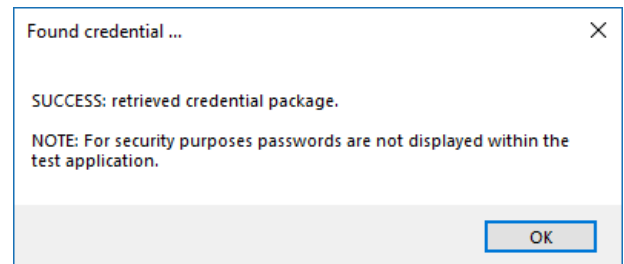
In either case, if the test succeeds but the results do not match what is expected, it's important to make note of the inputs which led to those results and verify permissions and access to credentials within the password vault.



When the search does yield one or more matching credentials, the test does allow for one additional level of verification by allowing a tester to retrieve a specific credential as would occur if it were selected for injection within the console. The tester simply clicks the **Retrieve Credential** button in the right column of the results list, and the integration then attempts to retrieve that credential on behalf of the supplied user.



The test displays the result of the attempt to retrieve the credential, but for security reasons no password is ever displayed in clear text.



Note: Only credentials are retrieved; no actual passwords are retrieved or displayed. 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. <div style="border: 1px solid black; padding: 5px; background-color: #e6f2ff;">  Note: The Test Settings feature does NOT communicate with BeyondTrust PRA at any point. It 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. </div>

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 B Series 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.