Privileged Identity 5.5.5.x
Installation Guide
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to Install BeyondTrust Privileged Identity</td>
<td>4</td>
</tr>
<tr>
<td>Install Privileged Identity Prerequisites</td>
<td>5</td>
</tr>
<tr>
<td>Host System Requirements</td>
<td>7</td>
</tr>
<tr>
<td>Management Console and Deferred Processing Host Requirements</td>
<td>7</td>
</tr>
<tr>
<td>Web Application Host Requirements</td>
<td>8</td>
</tr>
<tr>
<td>Web Service Host Requirements</td>
<td>8</td>
</tr>
<tr>
<td>Database Requirements</td>
<td>10</td>
</tr>
<tr>
<td>Service Account Requirements</td>
<td>12</td>
</tr>
<tr>
<td>Port Requirements</td>
<td>14</td>
</tr>
<tr>
<td>Install Server Components</td>
<td>16</td>
</tr>
<tr>
<td>Use PowerShell to Install Required Server Components on Windows Server 2016 or 2012</td>
<td>16</td>
</tr>
<tr>
<td>Use the GUI to Install Required Server Components on Windows Server 2016 or 2012</td>
<td>17</td>
</tr>
<tr>
<td>Further Steps to Enable Remote COM Access</td>
<td>19</td>
</tr>
<tr>
<td>Configure SSL on IIS</td>
<td>21</td>
</tr>
<tr>
<td>Create an SSL Certificate</td>
<td>21</td>
</tr>
<tr>
<td>Configure the Web App to Use Your Certificate</td>
<td>25</td>
</tr>
<tr>
<td>Managed System Requirements</td>
<td>27</td>
</tr>
<tr>
<td>Managed Database Requirements</td>
<td>29</td>
</tr>
<tr>
<td>Install Privileged Identity Software</td>
<td>32</td>
</tr>
<tr>
<td>Install the Management Console</td>
<td>32</td>
</tr>
<tr>
<td>Set Up the Program Database</td>
<td>37</td>
</tr>
<tr>
<td>Register the Privileged Identity Instance</td>
<td>44</td>
</tr>
<tr>
<td>Install the Web Application</td>
<td>45</td>
</tr>
<tr>
<td>Install the Web Service</td>
<td>49</td>
</tr>
<tr>
<td>Final Setup Steps</td>
<td>55</td>
</tr>
<tr>
<td>Enable Cross-Origin Resource Sharing</td>
<td>55</td>
</tr>
<tr>
<td>Connect to High-Availability and Cloud Databases</td>
<td>56</td>
</tr>
<tr>
<td>Mirrored Databases</td>
<td>56</td>
</tr>
<tr>
<td>SQL AlwaysOn</td>
<td>57</td>
</tr>
<tr>
<td>Azure SQL</td>
<td>58</td>
</tr>
</tbody>
</table>
Configure Encryption Options ................................................................. 59
  Configure Software-Based Encryption .............................................. 59
  Configure Hardware-Based Encryption ............................................ 60
  Finish Configuring Encryption ......................................................... 61
Configure URL Redirects ........................................................................ 62
Configure SMTP Email Settings ............................................................ 63
  General Settings .............................................................................. 63
  Outgoing Server Settings ................................................................. 64
  S/MIME Settings ............................................................................. 65
  OAUTH2 Authentication Settings ....................................................... 66
Firewall Configuration ........................................................................... 67
SMTP Logging Settings .......................................................................... 68
Require SSL ............................................................................................ 69
Require User Certificates ....................................................................... 70
Enable Integrated Windows Authentication ........................................ 71
How to Install BeyondTrust Privileged Identity

BeyondTrust Privileged Identity helps you secure, manage, and administer credentials for privileged users and IT vendors. With Privileged Identity, you can discover, maintain, and rotate passwords for privileged accounts to improve security and compliance.

This guide covers everything you need to know to get Privileged Identity up and running, including prerequisites, software installation, and final setup steps.
Install Privileged Identity Prerequisites

In this section, we’ll cover the installation prerequisites for Privileged Identity. Based on your starting host system configuration, your actual installer experience may vary.

Recommended Knowledge

While we provide documentation and support to set up and configure Privileged Identity along with the various technologies it uses, product admins should have knowledge in the following areas:

- Microsoft SQL Server and all target databases
- IIS web server technologies
- Network administration
- System administration

Product Requirements

Privileged Identity is an n-tier product, with individual components operating exclusively of each other. While the different components can be hosted on a single system, we recommend that you distribute them across multiple systems as resources permit. When deployed across multiple hosts, loss of the management console would have no functional effect on the web app or web service, just as loss of the web app or web service would not affect the management console.

Privileged Identity requires a Microsoft SQL Server database to store program data. All component host servers should be patched, secured, and configured according to your corporate patching strategy to help ensure that the password store will not be compromised.

The primary components are:

- **Management Console** - The primary administrative interface for general configuration of the software

  For more information, see "Host System Requirements" on page 7.

- **Web Service** - The resource used by various components, including the web app, to perform programmatic access and management of the product
- **Web Application** - The primary user interface to retrieve managed credentials or establish sessions
- **Database** - The data store where managed passwords and most program configurations are stored

  **Note:** All software components require communication with the database.

Privileged Identity is supported in a physical, virtual (cloud), or physical-virtual mixed environment. If any components will be shared on a single host, then simply combine the requirements.

The database should be placed on a separate system to keep the encrypted data segregated from the encryption key.

Additional components include the following:

- **Deferred Processing Service** - Used with scheduled jobs and automatic retry options. Comes included in the download package.
- **Zone Processors** - Used to manage segregated, distributed, and untrusted networks.

  **Note:** Zone processors are a licensed feature of Privileged Identity.

- **Integration Components** - Additional connectors used by zone processors, remote web services, and web applications to integrate with email, help desk systems, syslog output, etc. Comes as IntegrationComponents.msi.

- **Cross-Platform Support Library** - Used to manage and discover non-Windows-based systems and devices (e.g., Linux, Unix, iOS) from zone processors. Comes included in the download package as CrossPlatformSupportLibrary.msi.

- **Email Server** - (Optional) Used to send email notifications. Configuration of an email server (including enabling SSL and establishing a certificate trust) is done outside Privileged Identity.

- **IIS Media Services** - (Optional) Used to stream recordings of sessions run through application launching. Comes included in the download package.

---

**IMPORTANT!**

A poorly configured virtual host can impede the software’s ability to work. Make sure that the recommended resource allotments for each component have been met, and if possible, allow for dynamic increases in memory and storage. Supported host virtualization platforms are Hyper-V Server 2012 R2 or later, VMware ESX, and VMware Workstation.
Host System Requirements

In this section, we'll cover the server requirements for the various component host systems.

Management Console and Deferred Processing Host Requirements

Platform Requirements

For a production install of Privileged Identity, you must have a Windows Server operating system. Windows Server 2012 R2 and above are supported. We recommend using the most current version of the Windows Server as the host platform. Privileged Identity is fully supported on a physical server or virtual machine. All service pack levels and editions are supported except where specifically noted.

Note:

- The Privileged Identity software is a 32-bit application that runs in a WOW64 environment on 64-bit systems.
- When managing COM on Windows 2000 target systems, there will be inconsistencies with Remote COM management interfaces, as your host operating system will not match. This is by Microsoft's security design.
- Take note of any components which will manage databases other than Microsoft SQL Server. This should include the management console as well as any deferred or zone processors. These components must have the most recent 32-bit OLE DB providers installed, typically available from the database provider.
- Make sure that Microsoft .NET Framework 4.5.2 or later is installed on the host system. We recommend using the latest version and service pack of .NET available from the 4.x family.

To install the most recent version of .NET, please see https://dotnet.microsoft.com/download.

- To integrate System Center Service Manager (SCSM) or System Center Operations Manager (SCOM), you must obtain the appropriate SDK binaries from the SCSM or SCOM installation directory and place them into the Privileged Identity installation or zone processor directory.
- A poorly configured virtual host can impede the software's ability to work. Make sure that the recommended resource allotments for each component have been met, and if possible, allow for dynamic increases in memory and storage. Supported host virtualization platforms are Hyper-V Server 2012 R2 or later, VMware ESX, and VMware Workstation.
- For lab environments, Windows 10 Professional 64-bit will suffice. Workstation-class operating systems are not supported in production environments.

Hardware Requirements

In addition to the requirements for the host system and database, the Privileged Identity software itself requires the following:

<table>
<thead>
<tr>
<th>Minimum Requirements</th>
<th>Suggested Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 R2 or later</td>
<td>The most current version of Windows Server</td>
</tr>
<tr>
<td>2GB of RAM</td>
<td>2GB of RAM for the software</td>
</tr>
<tr>
<td></td>
<td>4GB of RAM for the program database</td>
</tr>
</tbody>
</table>
**Minimum Requirements**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Suggested Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>~1GB of hard drive space to install</td>
<td>4GB of hard drive space to install and for local log files</td>
</tr>
<tr>
<td>Intel or AMD multi-core system</td>
<td>Intel or AMD processors with 4 or more CPU cores</td>
</tr>
<tr>
<td>Microsoft .NET Framework 4.5.2 or later</td>
<td>The most recent version of Microsoft .NET Framework from the 4.x family</td>
</tr>
<tr>
<td>Windows Management Framework 4.0 or later</td>
<td>The more recent version of Windows Management Framework</td>
</tr>
</tbody>
</table>

*Note: This does not include space required by log files, which are enabled by default and can consume large amounts of space over time.*


**Required Server Components**

You may need to install IIS on the management console host system and/or zone processor host systems, even if the web app will not be installed on those systems. Some IIS components are required to install the web app to a remote server and to manage remote IIS installations.

For more information about installing these required components, please see "Install Server Components" on page 16.

**Web Application Host Requirements**

The Privileged Identity web application provides access to managed credentials and other functionality using a web browser.

Because the web app is deployed as an IIS web application, you must install certain components of IIS on the host server. The web app requires IIS 8.5 or above, which, in turn, requires Windows Server 2012 R2 or above (we recommend using the most current version of Windows Server).

*Note: These requirements differ from those for the web service host.*

For more information, see "Install Server Components" on page 16.

The management console can push the web application to the same system or to a remote web server.

**Web Service Host Requirements**

The Privileged Identity web service is required by the web app, PowerShell, and session recording.

You must install certain components of IIS and application server roles on the host server. The web service requires IIS 8.5 or above, which, in turn, requires Windows Server 2012 R2 or above (we recommend using the most current version of Windows Server).

*Note: These requirements differ from those for the web app host.*
Make sure that Microsoft .NET Framework 4.5.2 or later is installed on the host system. We recommend using the latest version and service pack of .NET available from the 4.x family.

Also, a valid SSL certificate is highly recommended. Certificate and authentication configuration affect browser support.

**IMPORTANT!**

*If the web service is hosted at a different URL than the web app, CORS support (Cross-Origin Resource Sharing) must also be enabled in the web service's web.config file, and additional browser configuration may be required.*
Database Requirements

A Microsoft SQL database is required at the time of Privileged Identity installation and serves as the software's storage and configuration data store. This database stores management sets, system information, account information, stored passwords, event sinks, answer files, email files, and more.

You can implement a new instance on an existing database server, or you can set up a new database server altogether. We recommend not sharing database instances with other applications. We also recommend placing the database on a system separate from other Privileged Identity components to keep the encrypted data segregated from the encryption key.

Supported Versions for Production Environments

We recommend using the most current version of SQL Server available. If you must use an older version, any version from SQL Server 2012 or later is supported, including Azure SQL Database.

Note: Microsoft Azure SQL Database requires Privileged Identity to use the latest version of the Microsoft SQL Native Client (not supplied with product download).

Both 32-bit and 64-bit versions of Microsoft SQL Server are supported. Standard and Enterprise editions are supported.

Supported Versions for Test Environments Only

Microsoft SQL Server Express is a lightweight version of SQL Server that is available for free download from the Microsoft web site. SQL Express should be used for testing scenarios only. If you must use SQL Server Express for testing, any version from SQL Server 2012 Express or later is supported.

IMPORTANT!

Using SQL Express will impact performance, scalability, and high availability options, and disaster recovery options.

SQL Express configures itself to a random port number during installation. The port number is required to complete the installation of Privileged Identity. See Microsoft documentation for details.

Components Which Need Data Store Access

The following components require access to the database:

- Management Console
- Web Service
- Deferred Processor / Zone Processors

Data Store Authentication

The following methods may be used to authenticate to a Microsoft SQL database:

- Local SQL Account Authentication / Explicit Database Authentication
- Integrated Windows Authentication
Whichever method you configure in the management console at the time of component deployment will also be the method used by the web application, the deferred processor, and the zone processors.

While you may use either authentication method, we recommend Integrated Windows Authentication, as this allows for additional logging and permits more granular control over who can access stored information. If you choose SQL authentication instead, all access to the database server happens in the context of the SQL account rather than the account of the user performing the action.

Whichever method you choose, you must provide the SQL account, the Windows user account, or the Windows group with access to the Privileged Identity database.

**Data Store Permissions**

If using a dedicated instance of Microsoft SQL, grant:

- **SYSADMIN** = server role
  
  or

- **Control Server** = database server right

This allows granted users the rights to perform all actions within that instance of SQL, including creating databases, storing procedures, and using all other features in the main application, as well as performing backup and restoration.

This allows granted users the rights to perform all actions with that instance of SQL, including:

- Creating databases
- Storing procedures
- Using all features in the main application
- Performing backup and restoration

If you don’t want or are not permitted to grant **SYSADMIN** or **Control Server** to the SQL instance, then the database administrator must create the Privileged Identity database beforehand. The SQL account or Windows user/group must be granted the following roles/rights over the Privileged Identity database:

- **DBO** = user role
  
  or

- **db_datareader** = user role
- **db_datawriter** = user role
- **db_dldadmin** = user role
- **EXECUTE** = database permission
- **CREATE TABLE** = database permission required during install/upgrade
- **CREATE VIEW** = database permission required during install/upgrade

If you are using explicit database permissions rather than granting **SYSADMIN** or **DBO**, then once the account has been granted the db_ roles above, you must grant the remaining permissions using SQL statements such as **GRANT EXECUTE TO username**.

Additionally, Privileged Identity can use the performance recommendations made by SQL Server for defragmentation, auto-index creation, etc. To do so, the SQL account or Windows user/group must be granted **View Server State** on the host SQL server.

**Note:** If **View Server State** is not granted, a database administrator must regularly tune the product database, or performance will decrease over time.
Service Account Requirements

The service accounts that run each component have different requirements in terms of security and access within your environment and within the Privileged Identity software. Because of these different requirements, we recommend using multiple service accounts to separately perform the specific functions of the software, minimizing permissions granted for any component service account. Alternatively, you may use the same service account for all functions of the software if doing so better meets your business needs.

Web Service Identity

Privileged Identity uses a COM application for its interactions from the web service server to the application database. This application requires the use of a privileged account. The account can be configured as a NetworkService if using explicit database authentication (SQL account), though it should be configured as a domain member when using Integrated Windows Authentication to the database. If using a named identity, the account should have the following rights and memberships:

- Administrator of the web server host system (required)
- Domain user (recommended when authenticating domain users or working with domain groups)
- Log on as a batch job
- DBO rights for the application database (if using integrated authentication)

**Note:**

*If a single implementation of Privileged Identity will manage multiple trusting domains and if you use Active Directory for user authentication or Integrated Windows Authentication to the database, then the COM identity must be a trusted user for the target domains. Otherwise, you must manually configure an authentication server entry with explicit credentials for that domain. The COM application must be configured to run as a user; this account can be automatically managed by Privileged Identity.*

Deferred Processor / Zone Processor Service Identity

Privileged Identity performs all scheduled jobs, such as password change jobs or password verification reports, by using a service on the management console host system or by using a standalone service called a zone processor. This application requires the use of a privileged account. The account should be a domain member (as applicable) and should have the following rights and memberships:

- Administrator of the management console host system
- Log on as a service
- DBO rights for the application database (if using integrated authentication); system admin of the database not required
- Administrative rights over target managed systems

**Note:** If the service account/interactive user account cannot be an administrator of the target systems, then you must configure an alternate admin account for use by the software. If possible, avoid the use of alternate admin accounts when managing COM and DCOM objects. Also avoid scheduled tasks, as these interfaces do not allow for impersonation.

For more information, please see the Privileged Identity Admin Guide at www.beyondtrust.com/docs/privileged-identity/documents/5-5-5/pi-admin-5-5-5.pdf.

---

**SALES:** www.beyondtrust.com/contact  **SUPPORT:** www.beyondtrust.com/support  **DOCUMENTATION:** www.beyondtrust.com/docs
**IMPORTANT!**

*The account used to run the deferred processing service cannot manage itself.* If you manage this account through a scheduled job, then any job being run by that processor at that time is stopped mid-process. This leaves the job in a locked and incomplete state, requiring an administrator to resolve the issue. This also causes all other scheduled jobs to stop running until an administrator manually starts the service.

An alternative to using a named service account for the scheduling service is to configure the service to run as *LocalSystem* or as a *Microsoft Managed Service Account* (MSA). This negates password management requirements for the service.

However, you must also grant permissions to the database for the computer account *(ComputerAccountName$)* and ensure that the computer account is seen as an administrator of all managed systems. If the computer account is added to a new group in Active Directory to provide these administrative rights, the computer must be restarted.

### Syslog Forwarder Service Identity

The syslog forwarder captures syslog or MSMQ UDP output and forwards the traffic via TCP or TCP with SSL to a target collector. This program requires a Windows service to run. The account can be a local user account or domain account and must be granted *Log on as a service*.

## Port Requirements

The following ports may be used by Privileged Identity. Actual port usage will vary based on the options used and the systems managed. The port direction defined below is relative to the Privileged Identity component.

> **Note:** The following ports are the standard ports for the various protocols. These ports may have been changed on the target systems. It is the Privileged Identity administrator’s responsibility to determine if any of the target ports have been changed and to reflect that changed port when password change jobs or account discovery jobs are performed.

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>TCP, outbound, SSH</td>
<td>Used to manage SSH-based devices.</td>
</tr>
<tr>
<td>23</td>
<td>TCP, outbound, Telnet</td>
<td>Used to manage non-Windows devices that support Telnet.</td>
</tr>
<tr>
<td>25/465/587</td>
<td>TCP, outbound, SMTP</td>
<td>Used to send email. Only required if email notifications will be sent from Privileged Identity.</td>
</tr>
<tr>
<td>80/443</td>
<td>TCP, inbound, HTTP/S</td>
<td>Used to access the web application and web service.</td>
</tr>
<tr>
<td>88</td>
<td>TCP/UDP, outbound, Kerberos</td>
<td>Used by the jump server when authenticating with Kerberos.</td>
</tr>
<tr>
<td>135 &amp; Ephemeral ports</td>
<td>TCP/UDP, outbound, RPC port mapper service</td>
<td>Used for most Windows COM/DCOM-based operations. The remote DCOM management port and ephemeral ports are typically provided by granting access to <code>DLLHOST.EXE</code> in the <code>%systemroot%\system32</code> directory. Ephemeral ports vary by target Windows operating systems.</td>
</tr>
<tr>
<td>161</td>
<td>TCP, outbound, SNMP</td>
<td>Used during system/network discovery operations and device management functions.</td>
</tr>
<tr>
<td>389/636</td>
<td>TCP, outbound, LDAP/LDAPS</td>
<td>Used for LDAP-compliant directories such as Active Directory.</td>
</tr>
<tr>
<td>443</td>
<td>TCP, outbound, HTTPS</td>
<td>Used for ESXi native management, as well as various cloud service providers and SAML/OAUTH authentication providers.</td>
</tr>
<tr>
<td>445</td>
<td>TCP, outbound, SMB</td>
<td>Used for Windows Server.</td>
</tr>
<tr>
<td>464</td>
<td>TCP/UDP, outbound, Kerberos</td>
<td>Used by the jump server when authenticating with Kerberos.</td>
</tr>
<tr>
<td>514</td>
<td>UDP, outbound, syslog</td>
<td>Used to communicate to logger systems such as ArcSight, QRadar, Splunk, syslog, etc.</td>
</tr>
<tr>
<td>623</td>
<td>UDP, outbound, IPMI</td>
<td>Used to manage lights-out devices such as Dell DRAC, HP iLO, etc.</td>
</tr>
<tr>
<td>1025</td>
<td>TCP, outbound, Teradata</td>
<td>Used to discover and manage Teradata databases.</td>
</tr>
<tr>
<td>1433</td>
<td>TCP, outbound, MS SQL Server</td>
<td>Used to connect product components to the Microsoft SQL Server data store.</td>
</tr>
<tr>
<td>1521</td>
<td>TCP, outbound, Oracle</td>
<td>Used to discover and manage Oracle databases.</td>
</tr>
<tr>
<td>3306</td>
<td>TCP, outbound, MySQL</td>
<td>Used to discover and manage MySQL databases.</td>
</tr>
</tbody>
</table>
Port | Direction                      | Description                                                                                                                                                                                                 |
-----|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
3389 | TCP, outbound and inbound,     | Used for remote connections to target servers (automatic sessions) as well as inbound to the application launch server.                                                                                      |
      | Remote Desktop Protocol (RDP)  |                                                                                                                                                                                                             |
Port 5000 | TCP, outbound, Sybase          | Used to discover and manage Sybase ASE databases.                                                                                                                                                    |
Port 5432 | TCP, outbound, PostgreSQL     | Used to discover and manage PostgreSQL databases.                                                                                                                                                          |
Port 50000 | TCP, outbound, DB2            | Used to discover IBM DB2 databases.                                                                                                                                                                          |

Other ports may be required depending on the application being managed. If your setup uses additional external items or processes, additional ports are required. Please refer to the following table for known port connection requirements:

<table>
<thead>
<tr>
<th>Application</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC Remedy</td>
<td>TCP/UDP, outbound, BMC_AR_Port</td>
</tr>
<tr>
<td>HP Service Manager</td>
<td>TCP, outbound, HPSM Port</td>
</tr>
<tr>
<td>Microsoft SharePoint Server</td>
<td>TCP, outbound, the SharePoint</td>
</tr>
<tr>
<td>Microsoft System Center Configuration</td>
<td>TCP, outbound, typically Microsoft</td>
</tr>
<tr>
<td>Manager</td>
<td>file and printer sharing or</td>
</tr>
<tr>
<td></td>
<td>remote management ports</td>
</tr>
<tr>
<td>Oracle WebLogic</td>
<td>TCP, outbound</td>
</tr>
<tr>
<td>IBM WebSphere</td>
<td>TCP, outbound</td>
</tr>
<tr>
<td>Others</td>
<td>Check your integration component</td>
</tr>
<tr>
<td></td>
<td>port requirements</td>
</tr>
</tbody>
</table>

Additional ports may be required based on target system configuration or Privileged Identity configuration. For example, an SSH target listening on port 5555 must accept connections from Privileged Identity, and Privileged Identity must communicate out on that port to the target. Similarly, if the web service or web application is on a non-default port for its HTTP/S configuration, the firewalls must be configured to allow communication on those ports.
Install Server Components

In this section, we'll detail how to install and configure required server components. Consider your network and determine where you would like which components of Privileged Identity to be installed. Then follow the steps below for each installation, including the components required to fulfill the server's purpose.

The installation routine initiates with a prerequisite checker to help you install the prerequisites. Following these directions allow you to install the prerequisites without going through the checker.

To enable Remote COM access, please follow the instructions below as well as "Further Steps to Enable Remote COM Access" on page 19.

Use PowerShell to Install Required Server Components on Windows Server 2016 or 2012

To enable Remote IIS Management using the PowerShell command line interface, enter the following line:

```
Install-WindowsFeature Web-Mgmt-Console
```

In the command above, Web-Mgmt is the component required to manage Windows Server.

To enable Remote COM Access using PowerShell, enter:

```
Install-WindowsFeature AS-Ent-Services
```

To enable Web App Hosting using PowerShell, enter:

```
```

In the command above, Web-Windows-Auth is an optional parameter which enables the web service to support Integrated Windows Authentication.

To enable Web Service Hosting using PowerShell, enter:

```
Install-WindowsFeature AS-Http-Activation,Web-Windows-Auth
```

In the command above, Web-Windows-Auth is an optional parameter which enables the web service to support Integrated Windows Authentication.

**Note:**

*If you want more than one feature on the same system, you may combine the parameters in one line. For example:*

```
```
Use the GUI to Install Required Server Components on Windows Server 2016 or 2012

To install server requirements using the user interface:

1. On each host server, open **Server Manager**.
2. From the dashboard, click **Add roles and features**.

3. On **Before You Begin**, click **Next**.

4. On **Installation Type**, select **Role-based or feature-based installation**. Click **Next**.
5. On Server Selection, select your host server (or remote host server if managing a core installation) and click Next.

6. On Server Roles, select components to install based on how this server will be used.
   
   a. To enable Remote IIS Management and/or Web App Hosting, select Web Server (IIS).
   
   b. To enable Remote COM Access and/or Web Service Hosting, select Application Server.

   **Note:** If any of the above selections prompts you to add required features or services, do so.

7. Click Next.

8. On Features, click Next.

9. On Web Server Role (IIS), click Next.

10. On Role Services, select components to install based on how this server will be used.

    a. For Remote IIS Management, select Management Tools > IIS Management Console to manage IIS.


    c. For Web App Hosting, select:

       i. Common HTTP Features

          - Default Document
          - Static Content

       ii. Health and Diagnostics > HTTP Logging - (Optional) Used for troubleshooting

       iii. Performance > Static Content Compression

       iv. Security - Items are optional unless needed for your use case

          - Request Filtering - Allows you to restrict clients from making certain requests of the web server, such as limiting the size of requests or prohibiting access to certain URLs.

          - Client Certificate Mapping Authentication - Install if users are provisioned user certificates via Active Directory and if user-certificate-based authentication is required. This requires additional IIS configuration.
IIS Client Certificate Mapping Authentication - Install if users are provisioned user certificates and if mapping and certificate authentication should be performed in IIS rather than Active Directory. This requires additional IIS configuration.

IP and Domain Restrictions - Allows you to restrict source IP addresses and domain names from making requests of the web server.

URL Authorization - Allows you to restrict URLs and HTTP methods. This can increase security when used in conjunction with Integrated Windows Authentication.

Windows Authentication - Allows you to use Integrated Windows Authentication. This may require additional IIS configuration.

Management Tools > IIS Management Console

Note: If any of the above selections prompts you to add required features or services, do so.

11. Click Next.
12. On Application Server, click Next.
13. On Role Services, select components to install based on how this server will be used.
   b. For Web Service Hosting, select Windows Process Activation Service Support > HTTP Activation.

Note: If any of the above selections prompts you to add required features or services, do so.

14. Click Next.
15. On Confirmation, validate your selections, and optionally check Restart the destination server automatically if required.
16. Click Install.
17. After installation, you must restart any management consoles, deferred processors, and/or zone processors which were running when this process began. In Server Manager, also restart any stopped services.

Further Steps to Enable Remote COM Access

Privileged Identity requires Remote COM to discover and manage COM applications on remote systems, as well as to push the web app and the web service to remote systems.

In each of these cases, if Remote COM access is disabled on the target system, Privileged Identity will fail to perform the requested function and will log an error message.

In addition to enabling Remote COM access, you must ensure that your firewall permits the required traffic from the management console host system and zone processor host systems, or a similar error will occur.
<table>
<thead>
<tr>
<th>Rule</th>
<th>Program to Allow</th>
<th>Local Address</th>
<th>Remote Address</th>
<th>Protocol</th>
<th>Local Port</th>
<th>Remote Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM In</td>
<td>%systemroot%\system32\dllhost.exe</td>
<td>Any</td>
<td>Host IP</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>COM Port Mapper In</td>
<td>Any</td>
<td>Any</td>
<td>Host IP</td>
<td>TCP</td>
<td>135</td>
<td>Any</td>
</tr>
<tr>
<td>IIS In</td>
<td>%windir%\system32\inetsrvinetinfo.exe</td>
<td>Any</td>
<td>Host IP</td>
<td>Any</td>
<td>Ephemeral Ports</td>
<td>Any</td>
</tr>
<tr>
<td>COM Port Mapper In</td>
<td>Any</td>
<td>Any</td>
<td>Host IP</td>
<td>TCP</td>
<td>135</td>
<td>Any</td>
</tr>
</tbody>
</table>

Unless otherwise configured, communication begins on port 135 (RPC Port Mapper), which then negotiates a target ephemeral port through which Privileged Identity performs the requested work. Ephemeral port ranges are initially determined by Microsoft and are specific to the target Windows system. However, administrators may change these ports.

**Note:**

While the sections above detail how to enable Remote COM access using PowerShell or the GUI, you may also enable Remote COM by modifying the registry:

1. Run `regedit.exe`.
2. Select the subkey `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\COM3`.
3. Right-click the key `Remote Access Enabled`, and then select `Modify`.
4. In the `Edit DWORD Value` dialog, type `1`, and then click `OK`. 
Configure SSL on IIS

To encrypt traffic between the web server and the client browser, as well as to protect privileged passwords while they are in transit, you must configure SSL. Privileged Identity does not come with a pre-installed certificate. Rather, you must obtain a certificate from a public certificate authority, from an internal private certificate authority, or by using a free utility. You can also use a self-signed certificate or create one in IIS.

IMPORTANT!

Because SSL and early versions of TLS have certain security flaws, Microsoft recommends disabling SSL v3 and earlier and forcing the use of TLS 1.2.

For more information, please see this Microsoft article on disabling older versions of SSL and TLS at https://portal.msrc.microsoft.com/en-us/.

Create an SSL Certificate

1. On the web app host server, open Internet Information Services (IIS) Manager.
2. From the Connections pane, select your server node.
3. From the center pane, open Server Certificates.
   - To create a request to an external certificate authority, go to “External Certificate Authority” on page 21.
   - To create a request to an internal certificate authority, go to “In-House Certificate Authority” on page 23.
   - To create a self-signed certificate, go to “Self-Signed Certificate” on page 24.

External Certificate Authority

4. To create a certificate request to a third-party certificate authority, select Create Certificate Request from the Actions pane.
5. On the **Distinguished Name Properties** dialog, enter the **Common name** (the name of the server as entered in a browser). Fill in all fields, and then click **Next**.

6. Select the appropriate **Crytographic service provider**.

7. Set the **Bit length** to 2048 bits or higher to maintain compatibility with modern browser and systems.

8. Click **Next**.

9. Enter a name for the certificate request file, and then click **Finish**.
10. You must now send the certificate request file to the certificate authority. Once they have signed your certificate and returned it to you, select **Complete Certificate Request** from the **Actions** pane.


12. In **Friendly Name**, enter a name for easy identification.

13. Select **Web Hosting** as the certificate store, and then click **OK**.

14. The certificate is added to the **Server Certificates** list.

---

**In-House Certificate Authority**

- **IMPORTANT!**

  *Domain certificates are intended for use only with members of your internal Windows domain. Otherwise, you should use a certificate signed by a trusted root certificate authority.*

15. To create a certificate request to an in-house certificate authority, select **Create Domain Certificate** from the **Actions** pane.
16. On the Distinguished Name Properties dialog, enter the Common name (the name of the server as entered in a browser). Fill in all fields, and then click Next.

17. In Specify Online Certification Authority, enter or search for the path of a certificate authority in your Windows domain.

18. In Friendly name, enter a name for easy identification.

19. Click Finish.

20. The certificate is added to the Server Certificates list.

Self-Signed Certificate

IMPORTANT!

We do not recommend using a self-signed certificate in a production environment, as no other system will trust that certificate. Some components and systems do not work with untrusted certificates. A self-signed certificate must be distributed and installed on every system that will connect to the web app or web service. Otherwise, those components won't work and will generate a certificate error every time they're attempted. Instead, you should use a certificate signed by a trusted root certificate authority.

21. To create a self-signed certificate, select Create Self-Signed Certificate from the Actions pane.
22. Enter a name for easy identification, and then click **OK**.
23. The certificate is added to the **Server Certificates** list.

---

**Configure the Web App to Use Your Certificate**

1. On the web app host server, open **Internet Information Services (IIS) Manager**.
2. From the **Connections** pane, expand your server node, and then click **Sites**.
3. From the center pane, select the web site that hosts your Privileged Identity web app.
4. From the **Actions** pane, select **Bindings**.
5. From the **Site Bindings** dialog, click **Add**.

6. From the **Type** dropdown, select **https**.
7. From the **IP address** dropdown, select an IP or select **All Unassigned**.
8. You may leave **Port** as the default unless your network settings require you to change it.
9. Enter the **Host name** for your site.

**Note:** If you changed the **Port**, you must include it in the URL as `https://address:port_###/`. 
10. If you need to include a virtual domain as part of SSL negotiation, you may check **Require Server Name Identification**.
11. Select the appropriate certificate from the **SSL certificate** dropdown.
12. Click **OK**.
13. HTTPS binding is now appended to the web site. Click **Close**.
14. To require the web site to use SSL, select your site node from the **Connections** pane of the IIS manager.
15. In the **IIS** section of the center pane, open **SSL Settings**.

16. Select **Require SSL**, and then click **Apply**.
Managed System Requirements

In this section, we’ll cover many of the required services and expected configurations for target managed computers and devices. These requirements are generally the same: a credential able to connect and perform the desired management function.

IMPORTANT!

For credentials you’ll include in password rotation, you must know your password policy requirements. Otherwise, Privileged Identity could attempt to set passwords which don’t match the requirements of your policy, and if those changes succeed, problems will occur later.

For example, a device could allow the command to include special characters such as an @ symbol, but when the command is processed on the device, the @ symbol could be parsed as a string delimiter. This could either cause the entire command to fail or else report success but lock you out of management. Please be aware of your devices and their limitations.

Windows Requirements

- File and print services for Microsoft networks (installed and enabled by default)
- Server service (installed and enabled by default)
- Remote registry (optional; required to gather further system information such as MAC address, DCOM applications, etc.)

If you plan to use Privileged Identity to propagate, manage, or discover any of the following items, enable each respective requirement to support management via the native API:

- COM/MTS - Requires application server role with network COM access
- DCOM - Requires remote registry service
- IIS - Requires IIS management components, the application server role, and network COM access
- WMI - For SQL Server reporting services account; requires SCOM SDK binaries (from the SCOM server) to be placed in the Privileged Identity installation directory

Enabling remote access to COM and IIS requires additional configuration steps on the target systems. For more information, please see "Install Server Components" on page 16.

For information on port requirements, please see "Port Requirements" on page 14.

Linux/Unix/OSX Requirements

- Current SSH port - Required for password change and account enumeration
- Login password or SSH key - Required for the login account and possibly for the account being managed (operation-specific)
- Low-powered login account - (Optional) Used if root accounts are not allowed SSH access to the target system

Some distributions of Solaris, AIX, and other Linux/Unix distros may require password authentication to be enabled in the /etc/ssh/sshd config file. If this is required but not enabled, a password change job results in errors, saved in the log. To enable
password authentication, open the `/etc/ssh/sshd` config file and set **Password Authentication** to **Yes**. Then restart the SSH daemon. The restart method is distro-specific. Here are some examples of various restart commands:

- FreeBSD: `/etc/rc.d/sshd restart`
- Solaris: `svcadm restart network/ssh`
- Suse: `rcsshd restart`
- Ubuntu: `sudo /etc/init.d/sshd restart`
- Red Hat/Fedora/CentOS: `/etc/init.d/sshd restart OR service sshd restart`

**Cisco Requirements**

- Login account username and password
- Current password of the enabled account
- SSH or Telnet port if changed from the default

**IPMI Requirements**

- Root or admin-level login account username and password

**SSH/Telnet Devices Requirements**

Actual requirements vary based on target type and embedded operating system.

- Login account username and password or SSH key
- SSH port or Telnet port if changed from the default

For information about modifying the XML files used for SSH/Telnet targets, please see the [Privileged Identity Admin Guide (PDF)](https://www.beyondtrust.com/docs/privileged-identity/documents/5-5-5/pi-admin-5-5-5.pdf).

**Other Platform Requirements**

Other platforms have requirements specific to their implementation, configuration, and defined policies. Please see your target system's documentation for servicing requirements.
Managed Database Requirements

To manage a database through Privileged Identity, you must install the appropriate database provider on the host system that will perform the management tasks. Database providers may be downloaded from the database manufacturer.

**Note:** Privileged Identity requires 32-bit database providers. 64-bit providers are not supported.

The following databases require specific providers to allow you to manage their identities from the Privileged Identity host system.

**Note:** Windows comes with a Microsoft SQL Server provider. However, the SQL Server Native Client may be required for some specific configurations, requiring you to download the SQL Server Native Client provider. We recommend using the most current version of SQL Server Native Client. Using SQL Server Native Client 10 can result in undesirable behavior.

- PostgreSQL: Install the latest x86 provider from [www.postgresql.org/ftp/odbc/versions/msi/](http://www.postgresql.org/ftp/odbc/versions/msi/)

**Note:** The links above offer guidelines for installing the required providers. Please note that some vendors may require a login, a license agreement, or other prerequisites to download the provider. BeyondTrust is not responsible for third-party installations. You are responsible for all licensing and use restrictions surrounding these providers.

After you've installed the proper 32-bit OLE DB provider, it is available to Privileged Identity and is visible in the Add Target dialog when you add a new database target.

The rights required to change a target account's password vary from database to database; they also vary depending on the target account being changed. You may need other information, such as instance, service name, or port. Refer to your database provider's documentation for the most up-to-date description of rights required to change various identities. The sections below comprise a partial list of possible rights required for various databases.

**IBM DB2 Requirements**

For accounts associated with an IBM DB2 instance, the rights required depend on whether the database is hosted on Windows or Linux/Unix. DB2 has no local account store but instead references accounts from the host or related directories. If DB2 is hosted on Windows, follow the process for a typical Windows password change job. If DB2 is hosted on Linux/Unix, follow the process for a typical Linux/Unix password change job.

To enumerate accounts in a DB2 database instance (account store view), the login account requires:

- CONNECT TO DB
- GRANT SELECT on SYSIBM.SYSDBAUTH
Note: Privileged Identity can enumerate the local accounts associated with a DB2 instance. This process requires you to install the Microsoft-supplied OLE DB provider for DB2.

However, changing DB2 account passwords does not require a specialized provider, because DB2 uses the database host system’s local account store rather than providing its own internal account store.

Microsoft SQL Requirements

Microsoft SQL can leverage explicit SQL accounts or integrated authentication accounts. Accounts using integrated authentication are local computer accounts or accounts from a trusted domain. For either of these account types to manage account passwords within SQL, the following rights must be granted to the desired account or group:

- GRANT VIEW ANY DEFINITION
- GRANT CONTROL SERVER

The interactive login account and the deferred processing account require these rights to change passwords and enumerate accounts within the SQL database. Rights must be granted to a Windows user or group for Integrated Windows Authentication. The database instance name and port (if different from the default) are required.

Note: If the SYSADMIN right is granted, no other rights are required on the Microsoft SQL Server.

MySQL Requirements

A login account is required to configure a MySQL password change job. This login account must have sufficient rights to change the target account’s password. Assuming the login account can connect to the specified MySQL service and target database, the following global privilege must be granted to the login account:

- UPDATE

To enumerate the user accounts in a MySQL instance (account store view), the following global privilege must be granted to the login account for the appropriate database:

- SELECT

Oracle Requirements

A login account is required to configure an Oracle password change job. This login account must have sufficient rights to change the target account’s password. Assuming the login account can connect to the specified Oracle service (and instance, if applicable), the following right must be granted to the login account:

- ALTER USER

To enumerate the user accounts in an Oracle instance (account store view), the following right must be granted to the login account:

- SELECT ANY DICTIONARY

PostgreSQL Requirements

A login account is required to configure a PostgreSQL password change job. This login account must have sufficient rights to change the target account’s password. Assuming the login account can connect to the specified PostgreSQL service (and instance, if
applicable), the following right must be granted to the login account:

- **ALTER ROLE**

To enumerate the user accounts in a PostgreSQL instance (account store view), the following right must be granted to the login account:

- **SELECT**
Install Privileged Identity Software

In this section, we’ll cover how to install Privileged Identity and its components.

- "Install the Management Console" on page 32
- "Set Up the Program Database" on page 37
- "Register the Privileged Identity Instance" on page 44
- "Install the Web Application" on page 45
- "Install the Web Service" on page 49

Install the Management Console

Before you install the management console, make sure your server meets the prerequisites as defined in "Management Console and Deferred Processing Host Requirements" on page 7.

1. Launch the Privileged Identity installer. You should have received this from BeyondTrust Support or from the Support Portal.
2. In the prerequisites checker, select the components that will be installed on this machine.
3. From the dropdown, select the drive that Privileged Identity will be installed on.
4. Enter the information you will use to connect to the SQL server.
   - **DataSource String**: Enter the string used to connect to the data source in the form of Server\Instance.Port.
   - **Authentication**: Select whether to use integrated authentication or SQL Authentication.
   - **Username and Password**: Enter the credentials used to connect to the SQL server.
5. Click Next.

**Note**: Each component may use a different account to access the SQL server.
6. The checker runs a series of tests to let you know of any inadequate resources before beginning the installation process.

7. You can resolve any issues and then click **Re-run Prerequisite Checks**.

8. When ready, click **Next**.

9. On the welcome screen click **Next**.

10. Read the license agreement. If you agree, select **I accept the license agreement**, and then click **Next**.
11. Enter your name and organization name, and then click Next.

12. Select which features to install:
   - **BeyondTrust Privileged Identity** - (Required) Installs the Privileged Identity software. This is the only required element.
   - **PDF Encoder** - (Recommended) Allows you to turn compliance reports into PDF documents.
   - **PuTTY Terminal Emulator** - Installs the open source PuTTY software.
   - **RSA SecurID**
     - If RSA multi-factor authentication is required to access the management console but this machine will NOT host the web application, install this option.
     - If this machine will host the web application, leave this option unchecked. The RSA agent is installed automatically when the web application is installed.
   - **PeopleSoft** - Allows you to integrate Privileged Identity with a PeopleSoft database.

13. To change the installation location, click **Browse**.
14. To make sure you don't exceed your available disk space, click **Disk Cost**.
15. Click **Next**.
16. Choose which identity should run the Common Language Runtime (CLR) application. The default is **Network Service**.

The CLR COM identity provides Privileged Identity with network and local system access to various cloud services. Individual account stores (Azure, AWS, ESX, etc.) are configured with specific connection credentials when they are enrolled.

Options for the identity are:

- **Interactive User** - Use the same login information as the calling identity. This is an administrator-level account, as the calling identity will be either the admin running the console or the deferred processor service account. This option requires the least configuration but provides far more privileges than are required.

- **Network Service** - (Recommended) Use the system's NetworkService account. This does not require you to manage a password or grant additional rights, although in some cloud management cases, you may need to grant additional permissions on the file system.

- **Local Service** - Use the system's LocalService account. This does not require you to manage a password or grant additional rights, although in some cloud management cases, you may need to grant additional permissions on the file system. The LocalService account has many more rights than NetworkService.

- **This User** - Use a specified username and password. This user could be a local account that is configured to never authenticate to any other machine in the network (unlike NetworkService or LocalService), but it is another account whose credential you'll need to manage. You must grant this account **Logon as a batch** rights. In some cloud management cases, you may need to grant it additional permissions on the file system.

17. Click **Next**.

18. On **Ready to Install the Application**, click **Back** to make any needed changes or **Next** to complete the installation.
19. When the installation completes, you have the option to run Privileged Identity immediately.

The first time Privileged Identity launches, the program database setup wizard begins.

For more information, see "Set Up the Program Database" on page 37.
Set Up the Program Database

The first time you launch Privileged Identity, a setup wizard helps you configure various components of the Privileged Identity database. All of the component steps are optional except program data store configuration.

Note: If you need to run this setup wizard again, select Settings > Re-Run Setup Wizard in the management console.

1. On Database Setup, click Change Settings to create and/or connect to the database Privileged Identity will use for its primary data store.

2. Complete the fields in the Database Connection Information section.

Note: If you need to change this information later, select Settings > Data Store Configuration > Basic Configuration from the management console.
a. **Database access** - Select the database provider the Privileged Identity components should use when connecting to the data store.
   
i. **OLEDB Provider: SQL Server** - This is the default option.
   
ii. **ODBC Driver: SQL Server Native Client** - This is available if the SQL Server Native Client has been installed. You must use this option when connecting to mirrored databases, database availability groups, or Microsoft Azure SQL, or when using TLS v1.2 to encrypt database communications.

b. **Server name** - Enter the name of the server where the database should be created. This can be a simple name, fully qualified domain name, or IP address. If connecting to a named instance of SQL Server, enter the name as `ServerName\InstanceName`. If connecting to a custom port, enter the server name and port number separated by a comma, as `ServerName,###`. If you click the dropdown arrow, Privileged Identity populates a list with databases on the local network.

c. **Authentication information** - If you select Connect with Integrated Windows Authentication, then Privileged Identity connects to the data store using the service account or interactive management console account. This requires that these accounts (or groups they belong to) have been granted appropriate access to the database server and instance.

If you select Use database native authentication mode, then Privileged Identity uses a Microsoft SQL account to access the data store. This account is created and managed within the target SQL Server instance. Use this option when the database host is not trusted by the Privileged Identity component host (e.g., management console or web app host) or when you do not want to provision Active Directory groups or users to the database.

d. **Encrypt communication with database** - If this is deselected, traffic is encrypted using the database provider's encryption mode. Select this option to use SSL or TLS as controlled by your network administrator. The certificate must be trusted by the database server and all product component hosts.

e. **Add additional connection string parameters** - Supply additional connection string parameters specific to your database without having to write the whole connection string. This option is typically used when connecting to mirrored data sources or SQL AlwaysOn sources. When adding two or more parameters, separate them with a semicolon.

f. **Override settings - use custom connection string** - If you select this option, you can enter the connection string completely by hand. Only advanced database administrators should attempt this option.

g. **Test Connection** - Click this button to verify that you can make a connection to the specified database using the specified credentials. Once the test has completed, you can see the outcome in the **Status** section at the bottom of the dialog.

<i>For more information on connecting to mirrored databases, database availability groups, or Microsoft Azure SQL, please see "Connect to High-Availability and Cloud Databases" on page 56.</i>
3. After you've successfully connected to your database, complete the fields in the Database Settings section.
   a. **Name of the existing database to use** - Select a database instance to use, or click Manage Database Instances to view all found instances, to create a new database instance, or to delete an existing instance.

   **IMPORTANT!**
   Deleting a database instance removes it from the data store entirely, not just from this interface. A deleted database instance cannot be restored.

   b. **Use an explicit (non-default) schema** - This determines the context under which Privileged Identity will create database objects. We recommend checking this option and entering DBO in the field.

   If you leave this unchecked, then data is added under the context of your connected account. If this account is not in the sysadmin role, SQL Server creates a schema with your account name and creates all objects in that context. While this works for a single user or when using database native authentication, this option does not work well when using integrated security where connecting users are not sysadmin-level users.

   **IMPORTANT!**
   If you switch schemas, then any data already added will be removed. Be very careful about switching schemas if you've already begun using this database for this or other applications.

4. Optionally, enter Advanced Settings.
   a. **Set explicit connection limit** - This limits the number of connections made to the target database host. While this slows down job processing, it can improve stability when the database host is under-provisioned.

   b. **Maximum number of active DB connections during normal operations** - Set how many connections can be made at once.

   c. **Overwrite the default database timeout value** - Set how long Privileged Identity should wait for data to be returned from the database before the call times out. While 30 seconds should typically be enough, you may need to increase the timeout to handle high-latency, low-bandwidth links or while maintaining your database.

5. Click OK. Privileged Identity will now create all required views, stored procedures, and tables on the database. If no issues occur, the Database Setup dialog reappears, with Settings are Valid appearing in green.

6. Next, you'll perform Deferred Processor Setup. The deferred processor performs all scheduled actions within Privileged Identity.

   **Note:** If you need to change this information later, select Settings > Application Components from the management console, and then select Deferred Processor Service from the dropdown.
a. Supply a service account in the form of **Domain\Account Name**, and enter its password. This account must have local administrative rights, as well as the right to log in as a service on the local machine.

   ![Deferred Processor Setup](image)

   **Note:** If you don't have an account available at this time, click **Next** to skip this step.

   For more information about the deferred processor account requirements, see "Deferred Processor / Zone Processor Service Identity" on page 12.

b. Click **Install/Start Service**. Privileged Identity attempts to grant **Logon as a Service** to the account. If the process succeeds, you'll see **Service is Running** in green.

   If there are problems connecting to the database or granting rights, or if the account is not a local admin, the service fails to start. You can fix the issues now or after install.

c. Click **Next**.

7. To set up Privileged Identity to send email, enter **Email Setup**.

   **Note:** If you don't need Privileged Identity to send emails or if you don't have email server settings available at this time, click **Next** to skip this step.

   **Note:** If you need to change this information later, select **Settings > Email Settings** from the management console.
a. Click **Change Settings** to start the setup.

b. On the **General** tab, at a minimum, enter:
   
i. **Name** - The "from" name for the email.
   
ii. **Organization** - The name of your organization.

   iii. **Sender Email** - The "from" address for the email.

   iv. **Reply-to Email** - The address that replies should be sent to.
c. On the **Outgoing Server** tab, at a minimum, enter:

i. **Outgoing SMTP Server Name** - The DNS or IP address of the mail server.
ii. **Port** - The port through which to connect to the mail server.
iii. **Authentication Method** - The authentication your mail server is configured to use.
iv. **Use Authentication Credentials** - The username and password to connect to the mail server. If your mail server allows anonymous authentication, you can leave this unchecked.
v. **Use SSL Client Certificate Authentication** - Enter certificate information to secure mail traffic.

d. Click **Test Connection** to verify the settings, and click **Send Test Email** to check that emails send successfully.

e. Click **OK**. If there are no problems, you'll see **No problems detected with settings** in green.

---

**For details about SMTP email settings, please see "Configure SMTP Email Settings" on page 63.**

8. On **Setup Complete**, click **Edit Encryption Settings**.

> **Note:** If you need to change this information later, select **Settings > Encryption Settings** from the management console.

a. We highly recommend that you check **Use Encryption for Passwords in Database**.
b. Select the encryption type appropriate to your environment: software-based, FIPS 140-2, or PKCS #11. If you're unsure which to use, select Use software-based cryptography, with an Encryption Type of AES and a Key Length of 256 bit.

c. Click OK.

For details about encryption settings, please see "Configure Encryption Options" on page 59.

9. We recommend that you skip managing web application instances at this time, as not all web site options are enabled until you've completed registration. Furthermore, the web app requires the web service, which you have not yet configured.

For information on installing the web app, please see "Install the Web Application" on page 45.

10. Click Set Recovery Access Password to change the default password.

Note: You can change this password later by selecting Manage > View Stored Managed Passwords from the management console. If the password has not been set, you'll receive a prompt to set the password. Otherwise, from the Stored Passwords dialog, select Access > Change Recovery Access Password.

11. Click Finish.
Register the Privileged Identity Instance

**IMPORTANT!**
Register your Privileged Identity software on only the primary licensed console. Do not register any secondary or high-availability consoles. Doing so would lose all licensing information from the primary console, reverting it to demo mode.

**Note:** If you’re using a demo license, you don’t need to register at this time. Your demo software will be fully functional for 30 days and capable of managing up to 10 systems.

1. To register your software, select Help > Register from the management console.
2. Enter your name and your company name, and then enter your license key in the Serial Number field.
3. Alternatively, check Use Remote License.
   a. Select or add a remote license server, and then click OK.
   b. Click Connect As to select an alternate administrator account.
4. Click OK.
Install the Web Application

In BeyondTrust Privileged Identity, the web application is the primary method for accessing stored credentials, whether managed or static, as well as auditing access to those credentials. The web app also manages features such as the file store, the personal password store, and privilege escalation. In this section, we'll cover installation of the web app from the management console.

For information on the web app host requirements, please see "Web Application Host Requirements" on page 8.

1. In the management console, click Manage Web App in the left action pane.

2. On the Manage Web Application Instances dialog, click Install in the lower left corner.

3. On the Install Web Application dialog, select the target installation system.
   a. Local system is the computer you're currently working on.
   b. If you choose Remote System, enter the remote system's fully qualified domain name.
   c. Click Check System Compatibility. This checks that IIS and the file system are accessible on the target system, and that remote registry and Remote COM access are possible. Resolve any access errors before continuing.
   d. You will receive prompt to specify connection credentials. To use the currently logged in user account, click No. To specify a different account, click Yes, then enter the access credentials and click OK.

4. If the system compatibility check completes successfully, the Web Interface Files section is filled in automatically. If you need to change any of this information, here are the details:
a. **Install to target website** - All root web sites on the target server are listed here. Choose the root web site to host the web application.

b. **Web files destination path** - This is where the web application files will be copied. The path is resolved from IIS on the target server, which defaults to `%inetpub%wwwroot`.

5. If the URL to access the web application is different from the server name or virtual directory name, check **Explicit Site Address** and enter the appropriate URL. This URL might be different when using a load balancer or if the server name will be aliased in DNS.

6. Click **Install**.

7. You may receive a **COM Account Confirmation** warning. This appears if the COM account specified on the installation dialog is different from the currently logged in user. The warning asks you to be sure that the account specified has data store access. If it does not, the web app will fail to function until the access issue is resolved.

   If you are sure about the account information, click **Yes** to continue. Alternatively, click **No** to change to a different account.

8. When the web application install is complete, a success prompt appears. Click **OK**.

9. You will be prompted to launch the web application. Click **No** at this time.

10. Click **Close**. In the management console, the **Manage Web Application Instances** dialog will now be populated with a list of all known web applications.

### Supported Browsers

The web app has been tested with:

- Internet Explorer 11
- Microsoft Edge
- Google Chrome
- Mozilla Firefox
- Apple Safari
- Konqueror
- Opera

Following are known caveats when working with these browsers.

### Internet Explorer

- On Windows Servers with Internet Explorer Enhanced Security Mode enabled, the web app will not work unless its URL is added as a trusted site.
1. In Internet Explorer, select **Tools > Internet Options > Security**.
2. Select the **Trusted sites** icon.
3. Click **Sites**.
4. Add your web app URL to the list (e.g., https://server.example.int).
5. After closing the options, refresh your browser. The web app should now appear.

   - CORS support is available only on Internet Explorer 10 or later. To enable CORS, you may need to set this option:
     1. In Internet Explorer, select **Tools > Internet Options > Security**.
     2. Select the appropriate internet zone.
     3. Click **Custom Level**.
     4. Under **Miscellaneous**, enable **Access data sources across domains**.

**Microsoft Edge, Konqueror, Opera**

- This browser does not support the ActiveX control needed to launch RDP sessions.
- This browser does not support the ClickOnce extension needed to support application launching.

**Google Chrome**

- The IE Tab extension is required to support the ActiveX control needed to launch RDP sessions. This is currently supported only by Chrome for Windows.
- Only Chrome for Windows supports the ClickOnce extension needed to support application launching.
- SSL certificates that do not include a properly formatted subject alternative name are shown as insecure sites. This causes the user extra prompts and will likely break access to the web service, required for web app functionality.
- For Chrome to support Integrated Windows Authentication in scenarios where cross-origin requests (CORS) must be used, you must launch Chrome with the following flags:

  ```
  --disable-web-security --user-data-dir=SOLEDIRECTORY
  ```

  **Note:** Chrome will display a security warning. You can ignore this warning.

**Mozilla Firefox**

- This browser does not support the ActiveX control needed to launch RDP sessions.
- Only Firefox for Windows supports the ClickOnce extension needed to support application launching.
- For Firefox to allow Integrated Windows Authentication, the operating system must be joined to a trusted domain, and the following configuration must be made to the browser's profile:
  - For Kerberos authentication: `network.negotiate-auth.trusted-uris`
  - If Kerberos ticket passing is required: `network.negotiate-auth.delegation-uris`
  - If NTLM authentication is allowed: `network.automatic-ntlm-auth.trusted-uris`

  For the Kerberos exchange, define the domain name. If your domain name were `example.int`, you would enter `.example.int` (notice the leading dot).

- When the web app and web service are on separate machines and work with cross-origin requests (CORS), Firefox may not function properly when using Integrated Windows Authentication.
Apple Safari

- This browser does not support the ActiveX control needed to launch RDP sessions.
- This browser does not support the ClickOnce extension needed to support application launching.
- CORS support is available only on Safari 9 or later.
Install the Web Service

You will now install the web service, which is required by the web app, PowerShell, and session recording.

For web service host requirements, please see "Web Service Host Requirements" on page 8. For web service account requirements, please see "Service Account Requirements" on page 12.

You must install the web service locally to its host; it cannot be pushed to a target system from the management console.

To install the web service on the same machine as the management console, click Manage Web App from the left action pane, and then click Install Web Service at the bottom of the Manage Web Application Instances dialog.

To install the web service on a separate host, copy and use the manual installer. This is found in the installation directory, typically C:\Program Files (x86)\Lieberman\Roulette\SupplementalInstallers\ERPMWebService.exe.

1. On the web service host system, launch the web service installer.
2. On the welcome page, click Next.
3. On COM+ object identity, choose an identity and then click Next:
   a. Network Service - Choose this when using native-authentication mode to connect to the database.
   b. Interactive User - (Not recommended) Choose this when you want the user calling the web service to pass their authentication token to the database. This works when using Integrated Windows Authentication but requires considerably more security configurations in the program data store.
   c. Specific User - (Recommended, default) Choose this when using Integrated Windows Authentication to the database or when you want to minimize rights granted to the COM application. This is the most compatible option. Supply the username as DomainName\Username.
4. On **Web Installation Type**, select the location in IIS to install the web service, and then click **Next**:
   
a. **Virtual Directory** - (Recommended, default) This installs the web service to a virtual directory named **ERPMWebService**, located under the parent web site you’ll select next. This is the safest option to choose for both security and configuration reasons.

   **Note:** If you choose **Site** and if there are multiple root web sites configured on the host, you will be required to choose a root site.

   b. **Site** - Use this option to install the web service to the root web site.

5. If you chose **Virtual Directory** on **Web Installation Type**, select a web site on **Parent Site**, and then click **Next**.

6. If you chose **Site** on **Web Installation Type**, configure site options on **Web Site Configuration**.
7. On **Authentication Type**, choose how to connect to the web service, and then click **Next**:

   a. **Anonymous Auth with SSL** - Use this when SSL is configured but Integrated Windows Authentication will not be used.

   b. **Anonymous Auth without SSL** - (Not recommended) Use this when neither SSL nor Integrated Windows Authentication will be used. Some features will not work with this configuration.

   c. **Integrated Auth with SSL** - Use this when SSL and Integrated Windows Authentication will be used.

   d. **Integrated Auth without SSL** - Use this when Integrated Windows Authentication will be used but SSL is NOT configured. Some features will not work with this configuration.

   e. **SSL with User Certificates** - Use this when users must supply a user-based certificate (smart card, biometrics, etc.) to authentication to the web site and web service. This will cause more overhead in the overall configuration and may cause problems with some features.

   **Note:** Only methods available to the target parent web site are displayed.

8. On **Destination Folder**, choose where to install the web service, and then click **Next**. The default location is **C:\inetpub\wwwroot\ERPMWebService**, which automatically grants all permissions required for proper hosting. If you change the location, a web administrator may have to make further configurations.
9. Click **Install**.

10. Click **Finish** when the installer completes its process. The web service web page and tester will now open.

11. On the **Web Service Tester** make note of the **Web Service REST URI**; this is required for web app configuration.

12. If the web service and web app are installed on the same host, the web service requires no further configuration.

13. On the **Manage Web Application Instances** dialog, select the web app, then click **Edit**.

14. When prompted to confirm settings overwrite, click **Yes**.
15. On the App Options tab, find **Web service URI for REST web service endpoint** at the lower right of the dialog. Paste in the web service REST URI.

**Note:** If you have installed the web service on the same machine as the web app using the default settings, the web service REST URI is virtually the same as the web app URL.

For example, let’s say your server uses SSL on port 443 and your SSL certificate uses the fully qualified domain name of the server (`server.example.int`). The web service adds onto that (`/erpmwebservice/authservice.svc/REST`), making the URI `https://server.example.int/erpmwebservice/authservice.svc/REST`.

If you were behind a load balancer and the name of the load balanced cluster was `securestore.example.com`, the web service URI would be `https://securestore.example.com/erpmwebservice/authservice.svc/REST`.

16. Click **Test Connection** to verify the settings.

17. Click **OK**. When prompted that the settings have updated, click **OK** again.

**IMPORTANT!**

If you installed to a virtual directory, the install process creates a virtual directory called **ERPMWebService**. This directory inherits the authentication settings, SSL settings, and other settings from the parent web site. If the parent site is configured to use anonymous authentication and the web service installer is configured to use Integrated Windows Authentication, the virtual directory is created with faulty settings. To correct this, you must open IIS and reconfigure the authentication settings after install.
IMPORTANT!

If you install the web service on a machine that is NOT also hosting the web app, you must export the web app settings from the management console and import them onto the web service host. Otherwise, the web service will fail to load. To export the settings from the management console:

1. Click **Manage Web App** from the left action pane.
2. Select the desired web application instance from the list.
3. From the top tools menu, select **Advanced > Export web app registry config**. This exports a regedit file, which you'll save locally.
4. You will be prompted to generate the file for 64-bit Windows. Click **Yes**.
5. Copy the registry export to the target web service host and double-click the file to import it.

These steps provide the web service with the necessary information to connect to the data store, the hardware security module, the encryption key, and other settings. Any time these settings change on the web app host, you must repeat these steps.

IMPORTANT!

If the web service and web app have different host systems, and if the systems are accessed through different URLs (specifically the protocol, server name, or port), your web browser will block access to the web service, causing processes to malfunction.

To resolve this, enable cross-origin resource sharing (CORS). After you've installed the web service, open **web.config** and set **EnableCORS** to **true**.

Your specific browser may require additional configuration and may not work in all configurations. Please refer to your browser's documentation for more information on enabling CORS support.

Please see "Final Setup Steps" on page 55 for additional steps and verifications.
Final Setup Steps

After the web app and web service are installed, you may need to take additional steps, depending on the options enabled or desired. This section contains details about further setup steps you may need to perform.

- CORS Support - "Enable Cross-Origin Resource Sharing" on page 55
- Databases - "Connect to High-Availability and Cloud Databases" on page 56
- Encryption - "Configure Encryption Options" on page 59
- Redirects - "Configure URL Redirects" on page 62
- SMTP - "Configure SMTP Email Settings" on page 63
- SSL - "Require SSL" on page 69
- User Certificates - "Require User Certificates" on page 70
- Windows Authentication - "Enable Integrated Windows Authentication" on page 71

Enable Cross-Origin Resource Sharing

If the web service and web app have different host systems, and if the systems are accessed through different URLs (specifically the protocol, server name, or port), your web browser will block access to the web service, causing processes to malfunction.

To resolve this, enable cross-origin resource sharing (CORS). After installing the web service, open its web.config file (typically found at C:\Program Files (x86)\Lieberman\Roulette\ERPMWebService\web.config), and set EnableCORS to true.

**CORSDomain** controls the source domain allowed for CORS support. The initial value is an asterisk (*), which allows references from any web server. To limit communication to a particular domain, change the asterisk to your domain name. For example:

```xml
<add key="CORSDomain" value="example.int" />
```

This example sets Access-Control-Allow-Origin to example.int. Requests from servers in other domains will not be allowed.

**Note:** Only one **CORSDomain** value can be specified at a time.

**Note:** Your browser may require additional configuration. CORS may not work in all configurations.

About CORS

CORS is defined in [RFC6454](https://www.rfc-editor.org/info/rfc6454). This specification defines that a resource is considered the same origin if it uses the same scheme (protocol), host, and port. If your web app and web service are installed on the same host, are both accessed by HTTPS, and both use the same default port (443), they are considered to be of the same origin, and your browser will not block communication to either component. If any of these elements is different, the browser blocks communication to the web service from the web app, which prevents operations such as password retrieval.

Controls for browser behavior surrounding CORS vary by browser.

**For more information, please see “Supported Browsers” on page 46, and refer to your browser's documentation on enabling CORS support.**
Connect to High-Availability and Cloud Databases

For high availability and redundancy, you can set up the Privileged Identity program database using mirrored databases, database availability groups (SQL AlwaysOn), or Azure SQL.

To access database settings, select Settings > Data Store Configuration > Basic Configuration from the management console. If you change any database settings, you must restart the deferred processor. You must also update the web app configuration:

1. Select Manage Web App from the left action pane of the management console.
2. Right-click the web site instance.
3. Select Replace instance options with default web application options.

The sections below give details for setting up specific database configurations. For instructions about fields that are non-specific to these configurations, please see “Set Up the Program Database” on page 37.

Mirrored Databases

To connect to a mirrored instance of Microsoft SQL Server, make the following specific configurations:

- **Database access** - Select ODBC Driver: SQL Server Native Client. You must also manually install this provider on all web app servers, web service servers, and deferred and zone processor hosts. Otherwise, they will be unable to connect to the database.
- **Server name** - Enter the name of the primary (currently active) database partner.
- **Add additional connection string parameter** - Modify the following parameter with your mirrored server name:

  ```
  Failover_Partner=SECONDARY_SERVER_NAME;
  ```

- Click **Update** near the bottom of the screen, and review the updated connection string.
SQL AlwaysOn

To connect to a SQL Server database configured using SQL AlwaysOn, make the following specific configurations:

- **Database access** - Select **ODBC Driver: SQL Server Native Client**. You must also manually install this provider on all web app servers, web service servers, and deferred and zone processor hosts. Otherwise, they will be unable to connect to the database.

- **Server name** - Enter the name of the availability group listener (AGListener), prefaced with the protocol. For example:

  tcp:AGListenerName

- **Add additional connection string parameter** - Add the following connection string parameter:

  MultiSubnetFailover=True;

- **Click Update** near the bottom of the screen, and review the updated connection string.
Azure SQL

To connect to a SQL Server database configured using Azure SQL, make the following specific configurations:

- **Database access** - Select **ODBC Driver: SQL Server Native Client**. You must also manually install this provider on all web app servers, web service servers, and deferred and zone processor hosts. Otherwise, they will be unable to connect to the database.

- **Server name** - Enter the name of your Azure SQL instance, prefaced with the protocol. For example:

  tcp:servername.database.windows.net

- Click **Update** near the bottom of the screen, and review the updated connection string.
Configure Encryption Options

To configure Privileged Identity to work with software-based or hardware-based encryption, first select Settings > Encryption Settings from the management console.

**IMPORTANT!**

*When you change encryption settings, be sure to update the web app and web service settings. Otherwise, they will attempt to use invalid encryption mechanisms and will fail to access stored credentials and data.*

Configure Software-Based Encryption

When you select **Use software-based cryptography**, the generated key is unique to your installation. With an **Encryption Type** of AES, select a **Key Length** of 128, 192, or 256 bits.

To change the key, click **New Key**. After you confirm the change, the **Key Signature** updates. When recovering stored passwords, you can match the current key signature against the password's key signature to ensure that it was encrypted with the same key.

**IMPORTANT!**

*When you change encryption settings, passwords are decrypted and re-encrypted with the new key. You must manually update the web app settings to reflect the new encryption key.*

You may also test the validity of the current encryption key, export the current key, or import a different key. Exporting writes a registry file (.reg) with the encrypted key settings. You can later import these settings to the same system or to a different system by using the import feature or by double-clicking on the registry file.

**IMPORTANT!**

*Be careful when saving, importing, and exporting encryption keys. In the event of disaster recovery, if the key is lost, you cannot recover any passwords that have been encrypted with that key. Be sure to export your key and keep it in a secure location.*

**Software FIPS 140-2 Encryption Provider**

While the software encryption algorithms are FIPS algorithms, you may be required to use external FIPS 140-2 encryption modules if you are working with certain government organizations. The encryption code is the same for both methods; the FIPS 140-2 method simply uses the encryption procedures in a way that is compatible with the FIPS certification.
FIPS 140-2 requires you to use a certified stand-alone module. The Crypto++ library that Privileged Identity uses leverages the exact same cryptography code as a certified module. In the built-in case, the code is compiled into the software (not a certified use); in the certified case, the code is used through a call to an external, certified DLL. The certified use case is actually slightly less secure, as it is susceptible to replacement of the external DLL, whereas changing the built-in cryptography would require modifying the software itself, invalidating the digital signature.

To enable FIPS 140-2 certified encryption, download and install the module from your provider, which should contain the necessary add-on components, including the Crypto++ library. Then, check **Use FIPS 140-2 software provider if available**.

Choose what to do if the provider is not available. If you check **Only use FIPS 140-2 software provider**, the process will abort. Otherwise, the process will switch to the internal code.


Configure Hardware-Based Encryption

Hardware-based cryptography offloads the encryption process from the software and its host to an external device. You can use any hardware security module (HSM) that uses a PKCS #11 interface and supplies a 32-bit provider.

HSM technology is used by the government, military, and intelligence industries to protect against the security risks of conventional encryption software. In software-based solutions, even when keys are encrypted, software debuggers can potentially locate and access the encryption key, allowing critical data to be compromised. With an HSM, there is no record of keys stored in memory. Instead, keys are stored in a secure device, physically located inside an external piece of hardware.

To configure hardware-based encryption, select **Use Hardware Cryptography Module for hardware-based cryptography**, and follow the steps below:

1. Click the ellipses (…) to open the **PKCS #11 Interface to HCM Settings** dialog.
2. In **Interface library DLL path**, provide the path to the DLL that supports the HSM device.

   **Note:** When you install the hardware encryption device, it places a DLL on the host computer. This is required to interface with Privileged Identity.

3. If your device supports multi-threaded access, check **Initialize library for multi-threaded access** to improve performance.

4. Click **Load and Verify Library**. The **Library Description** field will be automatically filled.

5. Select a **Hardware Slot/Token**. The **Slot/Token Description** field will be automatically filled.

6. If the hardware devices requires a **PIN**, enter it now.

7. Under **Key and Encryption Method**, fill the fields required by your device.
   a. Select the **Key** from the dropdown. If no key exists and you have sufficient access, click **Create** to create a new key.
   b. Select the **Encryption Mechanism** from the dropdown.

8. Click **OK**.

---

**Finish Configuring Encryption**

If you check **Force change and clear any passwords which cannot be decrypted**, Privileged Identity examines all passwords in the password store and clears any passwords which cannot be decrypted using the current settings. This clears erroneous data from the database when the correct encryption key is unavailable. This is a single-use option. After this option is selected and the dialog is confirmed, the operation takes place; the next time this dialog is opened, the option will no longer be selected.

If you want to copy these settings to another system, enter the system's connection details, and then click **Push**.

**Note:** BeyondTrust is unable to provide support for your specific hardware security module (HSM). All support for your specific HSM must be handled by your HSM provider.
Configure URL Redirects

When you're installing or upgrading to a virtual directory, the virtual directory inherits the settings of the parent web site. Thus, if the parent web site is configured with a redirect, the virtual directory is also configured with a redirect. This can cause a redirect loop, in which the user can never connect to the web app or web service.

**Note:** URL redirects are not configured by default in IIS. They are used to redirect one address to another. For example, an attempt to reach the root web site using HTTP could redirect to the proper virtual directory with HTTPS.

To remove the redirect from the virtual directory:

1. On the host server, open Internet Information Services (IIS) Manager.
2. Expand your server node, then Sites, and then your web site.
3. Select your virtual directory. The default for the web app is PWCWeb, and the default for the web service is ERPMWebService.
4. From the center pane, open HTTP Redirect.
5. Clear all redirect options.
6. Click Apply.

**Note:** Other options to control switching from HTTP to HTTPS include:
- Using the Microsoft IIS URL Rewrite Module
- Creating a new default login page and configuring that page as the default document for the web site or virtual directory
Configure SMTP Email Settings

Privileged Identity can send email via SMTP for reporting and alerting purposes. You must have access to an SMTP server. To configure Privileged Identity to send emails, first select **Settings > Email Settings** from the management console.

**General Settings**

**Email Profile**

- **Profile Name** - You can create multiple email profiles, but only one can be used at a time.
- **Description** - Enter a short description of this profile.

**Sender Information**

This information is sent in the header of each email and appears to the recipient. Some email servers reject messages that don't have the proper address information for these fields.

- **Name** - Enter the "from" name for the email.
- **Organization** - Enter the name of your organization.
- **Sender Email** - Enter the "from" address for the email.
- **Reply-to Email** - Enter the address that replies should be sent to.
- **Read Receipts Email** - (Optional) Enter the address that read receipts should be sent to. A read receipt prompts the user to send a delivery status notification as soon as they open the email. If the recipient approves the receipt to be sent, their email client sends a reply to this address.
- **Return Receipt To Email** - (Optional) Enter the address that delivery receipts should be sent to. A delivery receipt requests the receiving mail server to send a delivery status notification as soon as it receives the email.

**Priority/Sensitivity/Importance**

- **Priority** - (Optional) Choose if emails should be sent with a status of **UNSPECIFIED**, **NORMAL**, **URGENT**, or **NON_URGENT**.
- **Sensitivity** - (Optional) Choose if emails should be sent as **UNSPECIFIED**, **PERSONAL**, **PRIVATE**, or **COMPANY_CONFIDENTIAL**.
- **Importance** - (Optional) Choose if emails should be sent with an importance of **UNSPECIFIED**, **HIGH**, **NORMAL**, or **LOW**.

**Advanced Message Settings**

**IMPORTANT!**

*Don't confuse this section with email subject lines. This section allows you to write custom MIME headers, which are added to the email before the body of the message appears. Do not enter any information in this section unless you need special headers and are comfortable writing MIME headers for email.*

- **Name** - Enter the attribute name to include in the message header.
- **Value** - Enter the attribute value to include in the message header.
Outgoing Server Settings

Outgoing SMTP Server Settings

- **Outgoing SMTP Server Name** - Enter the DNS or IP address of the mail server.
- **Port** - Set the port through which to connect to the mail server.
  Port 25 is standard for email, although it may be port 465 or 587 for SSL/TLS-encrypted email. Click Default to reset it to 25.
- **Server Timeout** - Set the number of seconds to wait for email to send.
- **Authentication Method** - Select the authentication method your mail server is configured to use. Incorrect method settings can prevent connection to a mail server even if the credentials are correct.
  - USER_PASSWORD - Basic username and password as defined below.
  - CRAMMD5 - Challenge-response authentication; protects passwords in transit.
  - NTLM - Challenge-response authentication; never sends a user password.
  - SASPLAIN - Challenge-response authentication; does not protect the password in transit.
  - KERBEROS - Kerberos authentication with the email server.
  - XOAUTH2 - OAuth authentication; requires configuration of the OAUTH2 Authentication tab.
- **SSL/TLS Channel Encryption** - If using SSL/TLS encryption, choose the option that your SMTP server is configured to use.
  - AUTOMATIC - Negotiate with the email server to find a supported SSL/TLS or plain-text method. Not all email servers support negotiation.
  - IMPLICIT - The mail server expects the initial connection to be already encrypted with SSL/TLS.
  - EXPLICIT - The mail server does not require the initial connection to be encrypted with SSL/TLS but may use SSL/TLS after the connection is initiated.
  - NONE - Use when automatic negotiation does not work and SSL/TLS is not configured on the email server.

Email Server Authentication

- **Use Authentication Credentials** - The username and password to connect to the mail server. If your mail server allows anonymous authentication, you can leave this unchecked.

Email Server SSL Settings

- **Use SSL Client Certificate Authentication** - Select this option if your SMTP server is configured to use SSL encryption. SSL encryption allows both logon credentials and data to be encrypted during the SMTP transaction. The server must be set up to use SSL encryption for this option to work.
- **User Certificate File** - Enter the path to the security certificate file.
- **User Authentication Certificate Store** - Enter the path to the certificate store, if configured.
- **User Certificate Password** - If required, enter the password to your certificate file.
- **Enable Cached Certificate** - Select if you want to allow the certificate information to be cached.

### Test Options
- **Test Connection** - Verify that you can connect to the SMTP server and that the server accepts the configured credentials. This option completes the handshake with the server, but it does not send mail.

  The program log records the transaction details:

  ```
  SetMailServer error: 11001, [11001] Host not found
  Failed to fill SMTP settings
  Failed to send email message error: Host not found.
  ```

- **Send Test Email** - Send a test email.

### S/MIME Settings

#### Sign Email
- **Sign Email** - Select this option to digitally sign outgoing email using Secure Multi-Purpose Internet Mail Extensions (S/MIME). This allows recipients to verify that the email was not tampered with.

  **Note:** If Privileged Identity cannot read the signing file you select, this check box will be automatically cleared the next time you open this dialog. To make sure the signing certificate file is valid, be sure to click **Verify** before closing these settings.

  - **Signing File** - Browse for a certificate located in the file store.
  - **Signing Certificate Store** - Choose from a list of certificates held in the certificate store.
  - **Signing Cert Password** - If applicable, enter the password used while exporting the certificate.
  - **Hash Algorithm** - Choose the algorithm used to prepare the message digest for signature.
  - **Enable Cached Certificate** - Select this option to allow the certificate to be cached in the program database; clear this option if the certificate should be loaded from the path specified above. You might want to enable this option if signing fails because you’re running components on different servers, which can’t access the required certificate locally.
  - **Attach Certificate to Email** - If you select this option, the certificate used to sign the message is encoded and included in the message signature.
  - **Verify** - Click to test that the email can be successfully signed.

#### Encrypt Email
- **Encryption File** - Select this option to encrypt outgoing email with the recipient's public key. The recipient must have the corresponding private key to decrypt the email.
IMPORTANT!

If you use Secure Multi-Purpose Internet Mail Extensions (S/MIME) to encrypt email, you must have an enterprise public key infrastructure (PKI). Only messages sent to recipients in your organization's address list can be encrypted. Recipients who do not have a certificate cannot read encrypted messages.

Note: If Privileged Identity cannot read the encryption file you select, this check box will be automatically cleared the next time you open this dialog. To make sure the encryption file is valid, be sure to click Verify before closing these settings.

- **Encryption File** - Browse for a certificate located in the file store.
- **Encryption Certificate Store** - Choose from a list of certificates held in the certificate store.
- **Encrypt Cert Password** - If applicable, enter the password used while exporting the certificate.
- **Encryption Algorithm** - Choose the algorithm used to encrypt the email.
- **Enable Cached Certificate** - Select this option to allow the certificate to be cached in the program database; clear this option if the certificate should be loaded from the path specified above. You might want to enable this option if signing fails because you’re running components on different servers, which can’t access the required certificate locally.
- **Verify** - Click to test that the email can be successfully encrypted.

**OAuth2 Authentication Settings**

- **ClientId** - Enter the ID of the OAuth client that was assigned when you registered Privileged Identity with the authorization server.
- **ClientSecret** - Enter the client secret that was created when you registered Privileged Identity.
- **ServerAuthURL** - Enter the URL of the authorization server.
- **ServerTokenURL** - Enter the URL used to obtain the access token.
- **AuthorizationScope** - (Optional) Enter the scope request or response parameter used during authorization. If the scope is not set, the authorization server will use the default access scope as determined by the server. To request a specific access scope, set this to a space-separated list of strings as defined by the authorization server.
- **AuthenticationValue** - Provide an authentication value if required by your authorization server.
- **Get OAuth Authentication** - Click to start the authorization process. A browser window opens to the OAuth authentication page you specified so that you can complete the authentication flow.
- **Sample Data** - Click to populate the configuration fields with sample data for a demo application.

  ! IMPORTANT!
  
  Clicking **Sample Data** will overwrite your form entries.

- **Hide Browser Response** - Click to suppress the confirmation pop-up that indicates successful authentication.

**Firewall Configuration**

- **Firewall Type** - If necessary, configure settings to connect to your SMTP server through a firewall. Select the type of firewall to connect through.
  - **None** - (Default) The client connects directly to the mail server.
  - **Tunnel** - Bypasses the local router and connects the email client directly to the email server.
  - **SOCK S4** - Basic proxy connection with no additional security that supports TCP.
  - **SOCK S5** - Basic proxy connection that combines TCP and UDP support and allows for domain name resolution (DNS).

- **Auto Detect Firewall** - Select this option if you want to automatically detect and use firewall system settings, if available.

- **Host** - (Optional) Enter the domain name or IP address of the firewall. If you provide the domain name, a DNS request will set this property to the corresponding address.

- **Port** - The TCP port of the firewall host is set automatically based on the selected firewall type, but you can edit it for non-default configurations.

- **User Name** - If the firewall requires authentication, enter a username.

- **Password** - If the firewall requires authentication, enter the password for the provided username.
SMTP Logging Settings

- **Enable Event Log Logging** - Select this option if you want Privileged Identity to write SMTP log events to the Windows event log. This can be helpful in troubleshooting problems with SMTP traffic.

- **Enable SMTP File Logging** - Select this option if you want Privileged Identity to write SMTP application log events to a text file. This can be helpful in troubleshooting problems with SMTP traffic.

File Logging Settings

- **Log File Name** - Enter the path to the .txt file where you want SMTP events to be logged.
Require SSL

When you're installing or upgrading to a virtual directory, the virtual directory inherits the settings of the parent web site. Thus, if the parent web site is not configured to require SSL, then your virtual directory does not require SSL.

To require SSL on your virtual directory:

1. On the host server, open Internet Information Services (IIS) Manager.
2. Expand your server node, then Sites, and then your web site.
3. Select your virtual directory. The default for the web app is PWCWeb, and the default for the web service is ERPMWebService.
4. From the center pane, open SSL Settings.

5. Select the check box Require SSL.
6. Click Apply.
Require User Certificates

When you're installing or upgrading to a virtual directory, the virtual directory inherits the settings of the parent web site. Thus, if the parent web site is not configured to require user certificates, then your virtual directory does not require user certificates.

To require user certificates on your virtual directory:

1. On the host server, open Internet Information Services (IIS) Manager.
2. Expand your server node, then Sites, and then your web site.
3. Select your virtual directory. The default for the web app is PWCWeb, and the default for the web service is ERPMWebService.
4. From the center pane, open SSL Settings.
5. Select the check box Require SSL.
6. Under Client certificates, select:
   - Accept - Allows users to pass a user certificate but also allows those who do not have a certificate. Select this option if some users require certificates but you are unsure if all have certificates.
   - Require - All users must supply a valid user certificate to access this site.
7. Click Apply.
Enable Integrated Windows Authentication

When you're installing or upgrading to a virtual directory, the virtual directory inherits the settings of the parent web site. Thus, if the parent web site is not configured to use Integrated Windows Authentication (or is misconfigured by also enabling another form of authentication), then your virtual directory inherits those same undesired settings.

To require Integrated Windows Authentication on your virtual directory:

1. On the host server, open Internet Information Services (IIS) Manager.
2. Expand your server node, then Sites, and then your web site.
3. Select your virtual directory. The default for the web app is PWCWeb, and the default for the web service is ERPMWebService.
4. From the IIS section of the center pane, open Authentication.

5. Right-click on Windows Authentication and select Enable.
6. If any other forms of authentication are enabled, right-click on those methods and disable them.

Your browsers may require additional configuration, as described below.

Internet Explorer

For Internet Explorer to allow Integrated Windows Authentication, the URL must be seen as being part of the local intranet rather than the internet or a trusted network. Internet Explorer will automatically recognize a location as being in the intranet zone only if its location is entered with its short name rather than its fully qualified domain name (FQDN).

If you access the web app and web service using their short names, your Integrated Windows Authentication configuration should be complete, SSL certificates permitting. If you access the web app and web service using their FQDNs, Internet Explorer will not treat these as intranet-zone items, and Integrated Windows Authentication will fail.

To allow Integrated Windows Authentication when using FQDNs, each user must have the web app and web service FQDNs added to the intranet zone in Internet Explorer. You may use a group policy to push out the proper settings.

To add the FQDNs to a single user's intranet zone:

2. Select the Local intranet icon.
3. Click Sites.
4. Add your web app and web service FQDNs to the list.
Google Chrome

Recent versions of Chrome support Integrated Windows Authentication when run from a Windows host, without further configuration required.

For Chrome to support Integrated Windows Authentication in scenarios where cross-origin requests (CORS) must be used, you must launch Chrome with the following flags:

`--disable-web-security --user-data-dir=SOMEDIRECTORY`

*Note: Chrome will display a security warning. You can ignore this warning.*

Mozilla Firefox

For Firefox to allow Integrated Windows Authentication, the operating system must be joined to a trusted domain, and the following configuration must be made to the browser's profile:

- For Kerberos authentication: `network.negotiate-auth.trusted-uris`
- If Kerberos ticket passing is required: `network.negotiate-auth.delegation-uris`
- If NTLM authentication is allowed: `network.automatic-ntlm-auth.trusted-uris`

For the Kerberos exchange, define the domain name. If your domain name were `example.int`, you would enter `.example.int` (notice the leading dot).

*Note: These settings may be lost between Firefox upgrades.*

*Note: When the web app and web service are on separate machines and work with cross-origin requests (CORS), Firefox may not function properly when using Integrated Windows Authentication.*