Securing Connections in MiContact Center

WHITEPAPER DOCUMENT

RELEASE 9.2

Mitel®

NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

TRADEMARKS

Securing Connections in MiContact Center 9.2
Release 9.2
October 2019
Document Version 9.2

PURPOSE

The purpose of this whitepaper is to provide information and guidance on configuring your MiContact Center 9.1 deployment in a web secure environment. This guide covers securing connectivity for clients of the MiContact Center Server, securing interservice and interserver communications, and securing connections to third-party applications such as Microsoft SQL Server and web services. This guide assumes that readers have working knowledge of security certificates and certificate authorities, their use, and relevant standards.

GETTING STARTED

The first step to utilizing TLS within your MiContact Center deployment is determining which certificate authority to use and obtaining the appropriate certificate. There are two types of certificate authorities, and the authority used will depend on your business needs. In addition, obtaining the appropriate type of certificate will also depend on your business needs, and the utilization of the certificate. Both certificate authorities and certificate types will be covered in this section.

CERTIFICATE AUTHORITIES

Organizations can obtain certificates from two types of Certificate Authorities (CAs):

Internal CAs deployed within an Enterprise Organization

External CAs provided by third-party vendors such as Verisign and GoDaddy

An Internal CA should be used when client connections within the organization are made by devices that are joined to the domain, or when devices are controlled by the IT organization and root certificates can be provided easily to these devices. The root certificate must be trusted to maintain the certification chain and to trust all certificates issued by the internal CA.

An External CA should be considered when the majority of connections are taking place over the WAN or Internet, where devices are not controlled by the organization, but must maintain secure connections. Most major certificate authorities are trusted by the major operating systems through the distribution of trusted root certificates by the operating system manufacturer.

A mix of both internal and external CAs can be used depending on the deployment. For instance, an internal connection to the MiContact Center Server may utilize a certificate from the internal CA chain, while connections to an external reverse proxy or firewall appliance may utilize a certificate issued by a third-party CA.

Self-signed certificates, while supported by Microsoft Windows are not supported by MiContact Center due to the inherent security risks associated with them. It is not recommended to use self-signed certificates within your environment without careful consideration of the security risks. Discussing these risks is beyond the scope of this whitepaper.

Choosing which certificate authority is right for you, or choosing to use a mix of certificate authorities, depends on the individual environment, deployment strategy, and infrastructure. MiContact Center supports certificates assigned from both internal and external certificate authorities.

TYPES OF CERTIFICATES

In addition to different types of Certificate Authorities (CAs), different types of certificates can be issued by these CAs.

- Single server certificates
- UCC/SAN Certificates
- Wildcard Certificates

Note: The various levels of Domain Validation (DV), Organization Validation (OV), and Extended Validation (EV) certificates are not discussed in this whitepaper.

As with the CA, the type of certificate that is used depends on the individual deployment, and the security strategy of the organization. MiContact Center works with all certificate types supported by Microsoft Windows.

A single-server certificate specifically defines the domain and/or subdomain to be secured. When utilized with MiContact Center, the single-server certificate must be issued to the FQDN of the MiContact Center Server.

A UCC/SAN certificate allows, at the time of issue, to specify Subject Alternate Names (SANs) within the certificate. This allows a single certificate to protect multiple subdomains. A UCC/SAN certificate should be used if you are utilizing multiple remote MiContact Center Servers acting in the Proxy Updater role.

Wildcard certificates allow for all subdomains within a domain to be protected. These certificates are typically denoted with an asterisk in the subdomain portion of the FQDN, for example *.mitel.com. This allows any server or service within the domain to be protected by a single certificate.

The strengths, weaknesses, and applications of each of these certificate types will not be discussed here. The type of certificate used for the MiContact Center is at the discretion of the organization.

Note: The certificate must be from a trusted certificate authority. If you apply a certificate that is not from a trusted certificate authority, MiCC services do not start.

SSL AND TSL PROTOCOLS

While the terms SSL and TLS are used interchangeably, they are in fact different protocol standards, and are individually maintained and implemented by the IETF. Most implementations of secure communication support both TLS and SSL, but typically will use one or the other. Due to possible security flaws within the SSL protocol, TLS is highly recommended to be forced in situations where both SSL or TLS could be used. When possible, this whitepaper will provide guidance on forcing connections to utilize TLS.

WINDOWS OR BASIC AUTHENTICATION MODELS

MiContact Center allows for two types of authentication models, Windows and Basic. When planning to implement your MiContact Center with secure communication channels, it is highly recommended, though not required, to use the Windows authentication scheme. This provides the highest level of authentication security within the contact center, in addition to providing a high level of authentication security when communicating with third-party applications.

CONFIGURING MICONTACT CENTER

The rest of this guide will discuss the configuration and implementation of a MiContact Center deployment with secure communication channels (where applicable). For demonstration purposes, all instructions and steps will show the use of a single server certificate, where you may be using a UCC/SAN or wildcard certificate. This guide assumes the certificate has been issued and imported to the Windows Certificates store for the Local Computer.

For information about requesting, assigning, and importing certificates in Windows Server, refer to the appropriate Microsoft documentation, and the documentation for your Certificate Authority.

CONFIGURING THE ENTERPRISE SERVER

To enable the MiContact Center Enterprise Server to use, and allow, secure communications, you must specify the FQDN of the Enterprise Server during installation. Ensure that the **I would like to use SSL** option is selected by default for new installations. If you are moving from an existing installation that is not using secure communication and want to enable the server, you must run the installation wizard and select the Repair Enterprise Server option.

In the IP Address screen for the MiContact Center Server installation, you must specify the FQDN of the MiContact Center server in the IP Address field and select the I would like to use SSL check box.

Figure 1, shows a sample installation that utilizes a server FQDN of SIQALAB01.MiCCCloud.com. The appropriate check box is selected for utilizing SSL in this deployment.

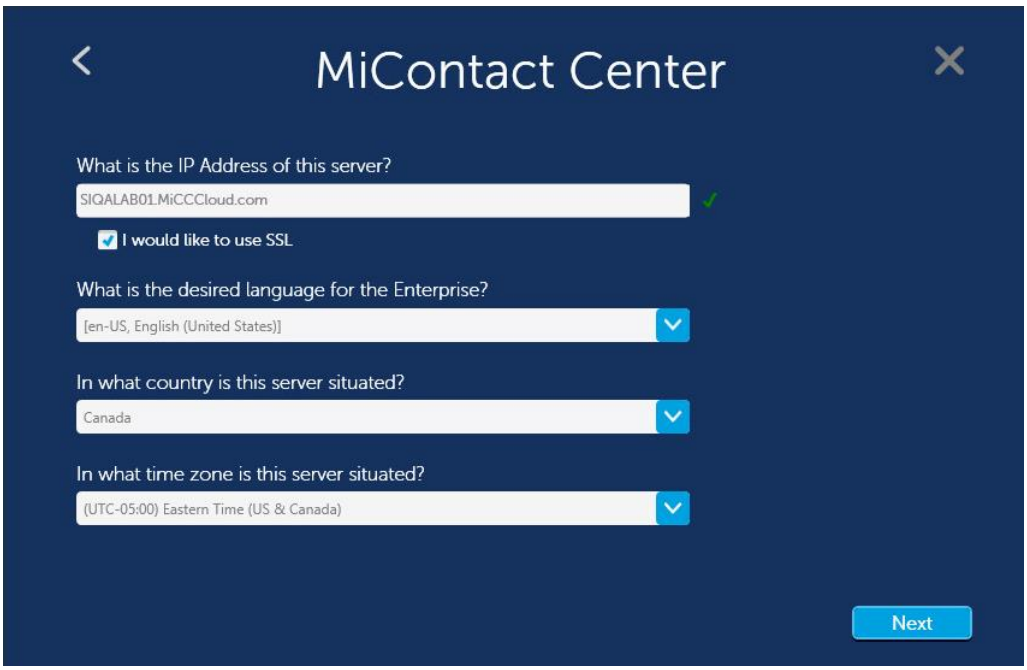


Figure 1: An installation configured for FQDN and SSL

If you select SSL, the SSL Configuration page opens. A set of prerequisite checks are executed to ensure that SSL is enabled in IIS and a valid certificate is bound to port 443. If the check fails, a list of valid certificates appears. You must select a valid certificate from the list and click **Recheck** to execute the prerequisite checks again before the SSL is enabled and the certificate is bound to port 443.

For instructions on manually configuring SSL and binding the certificate, see Configuring IIS for Secure Connections on page 6.

Figure 2: The SIQALAB01.MiCCCloud.com certificate has been used to configure IIS

The next step in the MiContact Center installation is to ensure you securely connect to the Microsoft SQL Server. To ensure this, connection, you must select to use Windows Authentication for the Microsoft SQL Server connection details. Figure 3 shows this in a sample installation.

In addition to specifying the Windows Authentication method for Microsoft SQL Server, you must ensure that the SQL Server is configured to allow only secure connections from SQL clients. The instructions for forcing secure connections to the Microsoft SQL Server are provided later in this document in the chapter CONFIGURING MICROSOFT SQL SERVER.

You must wait for the installation, or repair installation, to complete before continuing.

CONFIGURING IIS FOR SECURE CONNECTIONS MANUALLY

You must apply the Server Certificate to the website in Microsoft IIS manually before, during, or after the MiContact Center installation. The process includes adding HTTP binding on port 443, applying the certificate, and disabling the less secure SSL connections for TLS. If the installer does not create the binding, you must manually apply the Server Certificate to the website in Microsoft IIS manually before, during, or after the MiContact Center installation.

To add the HTTPS bindings:

1. Open the **Microsoft Internet Information Services (IIS) Manager**.
2. In the Connections page, expand the server.
3. Expand Sites.
4. Right-click **Default Website**
5. Select **Edit Bindings** (Figure 4).
6. In the Site Bindings window, click **Add**.
7. In the Type drop-down list, select HTTPS (Figure 5).
8. Leave IP address as All Unassigned to listen on all Network Interfaces.

9. Ensure Port is configured to 443.
10. From the SSL certificate drop-down list, select the appropriate certificate.
11. Click **OK**.
12. Click **Close** on the Site Bindings window (Figure 6).

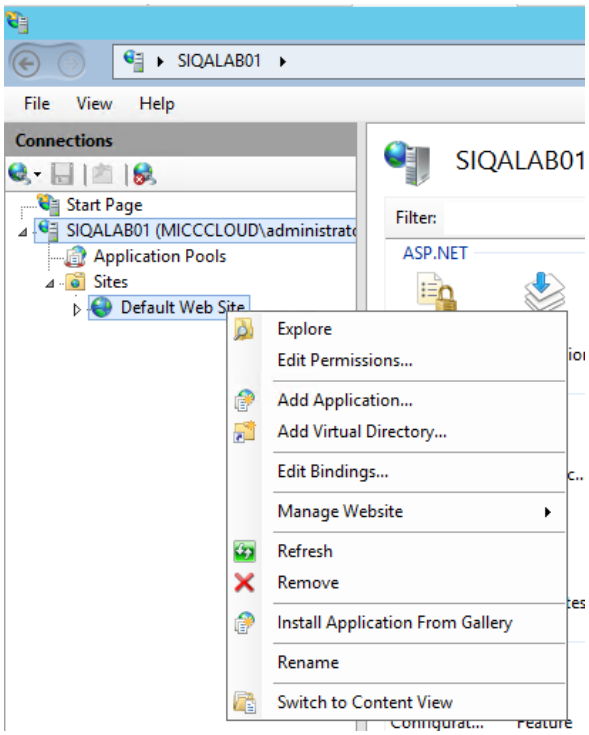


Figure 4: Edit bindings option in IIS Manager

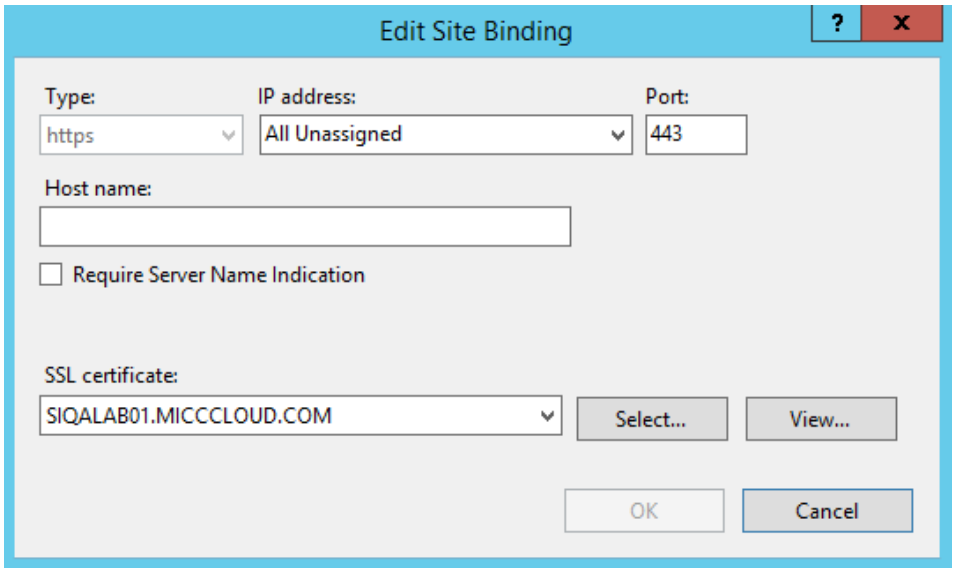


Figure 5: Sample site binding for HTTPS

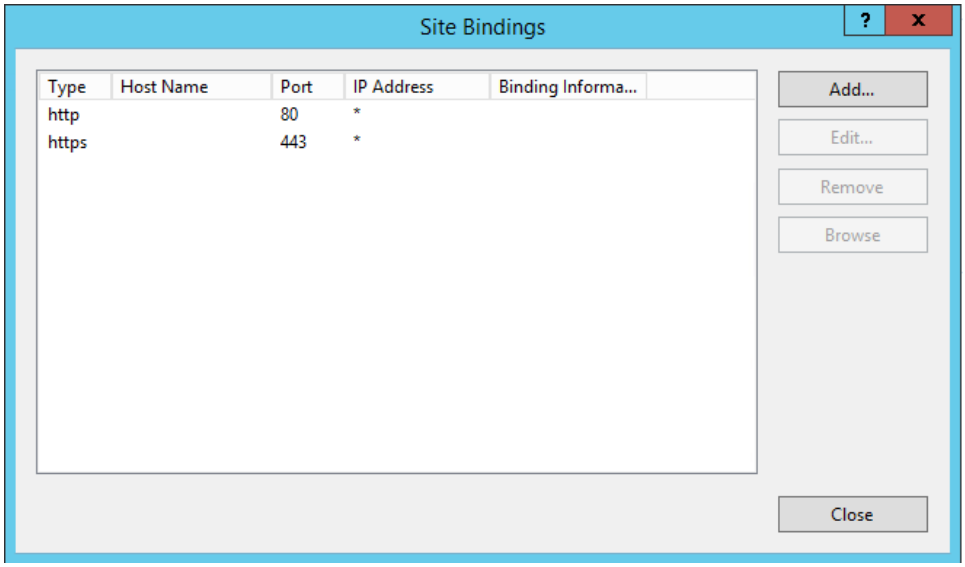


Figure 6: Sample completed site bindings configuration

After the site bindings are completed, you must perform an IIS Reset for the settings to take effect. To restart IIS:

1. In the Connections Pane of IIS Manager, right-click the server.
2. Click **Stop** and wait for the stop process to complete.
3. Right-click the server again.
4. Click **Start** and wait for the start process to complete.

Alternatively, to restart IIS, you can type IISRESET in an elevated command prompt, or restart the World Wide Web Publishing Service in the Windows Services control panel.

DISABLING HTTP CONNECTIONS

Optionally, you can disable connections over HTTP. Unless specifically required by your organization, Mitel recommends using HTTP connections for backwards compatibility with older MiContact Center applications and services.

To disable HTTP connections:

1. In the IIS Manager, right-click **Default Website**.
2. Click **Edit Bindings**.
3. Click the **HTTP binding** in the Site Bindings window.
4. Click **Remove**.
5. Perform an IIS Reset.

Note: For more information, refer to the steps regarding IIS Reset mentioned above.

In some scenarios, the HTTP connections for default websites might be required for other applications or URLs. In this situation, you can enable the SSL setting only for specific applications under the default websites in ISS.

To add SSL:

1. In IIS Manager > Default Website, select **Add Application** or **Add Virtual Directory**.
2. Select **SSL Settings** and click **Require SSL**.
3. Under Client Certificates, select **Ignore, Accept,** or **Require** based on your organization's requirements.

For information about the Require SSL settings, see the Microsoft documentation:

https://docs.microsoft.com/en-us/iis/configuration/system.webServer/security/access

FORCING CONNECTIONS TO UTILIZE TLS

Optionally, for additional security, you can disable connections over the less secure SSL protocol and enfore the use of the more secure TLS protocol. This may not be required as Windows Server 2012R2 as of December 2015 have SSL disabled and TLS forced by default.

CAUTION: The following steps entail modifications to the Windows registry. It is highly recommended that you back up registry values before proceeding with these steps. For information on how to back up your registry, see the following Microsoft documentation http://windows.microsoft.com/en-in/windows/back-up-registry

To enable TLS 1.2:

1. Open the **registry editor (regedit)**.
2. Browse to HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols.
3. Right click **Protocols** and click **New – Key**.
4. Name the key **TLS 1.2**.
5. Right click **TLS 1.2** and click **New –Key**.
6. Name the key Client.
7. Right click **TLS 1.2** and click **New – Key**.
8. Name the key Server.
9. Click the **Client key**.
10. In the right pane, right click and select **New – DWORD**.

11. Name the DWORD **DisabledByDefault**.
12. Ensure the value is set to **0**.
13. In the right pane, right click and select **NEW – DWORD**.
14. Name the DWORD **Enabled**.
15. Set the value to **1**.
16. Repeat steps 10 to 15 in the Server key.
17. For the settings to take effect, you must reboot the server.

SECURING AUDIO AND SIGNALLING

MiContact Center utilizes MiAUDIO to provide an audio channel to callers in both IVR Routing, and through the MiContact Center Softphone found within the Contact Center Client application. While the MiVoice Business Controller supports Secure Real-Time Transport Protocol (SRTP), SRTP implementations are not supported in ACD environments due to the resource overhead required for audio stream encryption.

Conversely, for signaling, MiContact Center uses the secure implementation of MiNET in both IVR Routing and Contact Center Softphone. There is no configuration required to enable the secure MiNET connectivity, as this is the default implementation when IVR Routing ports and Contact Center Softphones are configured with a Trusted Service Level in the MiVoice Business controller.

For additional information about SRTP and Secure MiNET see the MiVoice Business Systems Engineering Guidelines.

SECURING IVR DATA PROVIDERS

IVR Routing enables querying third- party data providers, such as web services and database providers. Depending on the nature of the data being queried or communicated between the IVR Routing server and the data provider, it may be required to configure secure connections to these endpoints.

Refer to the documentation for your data provider on how to communicate in a secure manner.

For web services, the web service URL must use the HTTPS prefix.

For ODBC data providers, you must configure the ODBC connection to utilize secure communication methods appropriate for the data provider. Refer to the data provider specification and documentation for information about configuring the ODBC driver accordingly.

CONFIGURING SECURE CLIENT CONNECTIONS

It is assumed for the purpose of this whitepaper that the client machines already trust the certificate authority of the certificate being used on the MiContact Center Server, either through the Active

Directory Enterprise Trust infrastructure or through the third-party root certificate available on the client machine (See Figure 7).

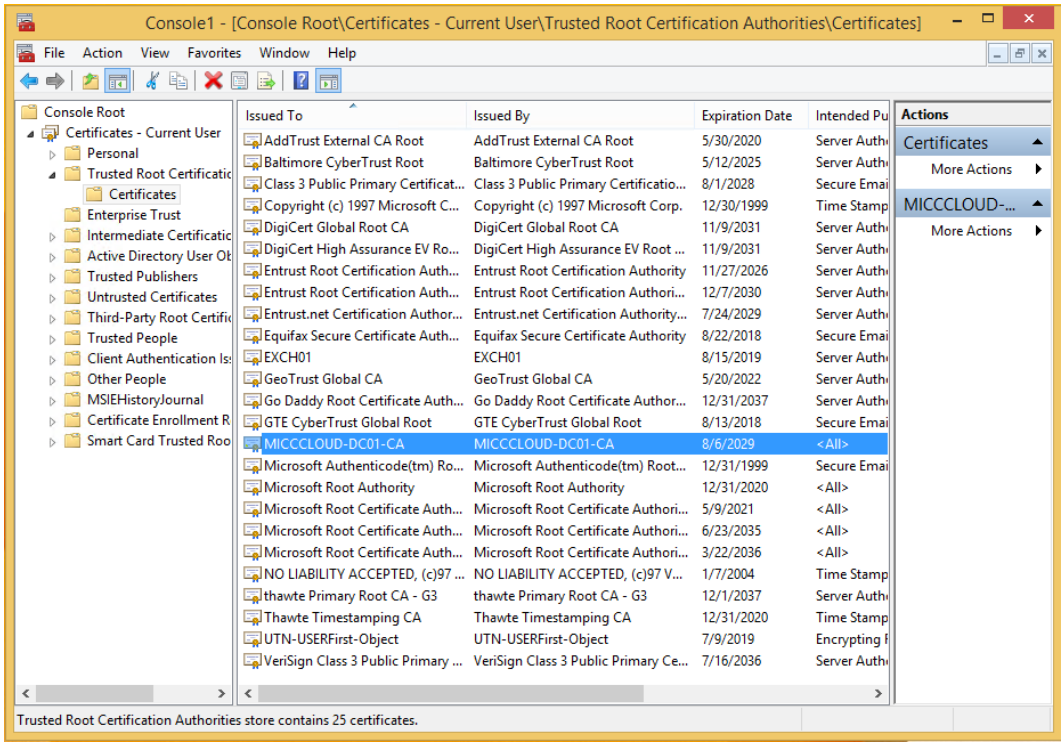


Figure 7: Enterprise Certificate Authority added as part of a domain joined client machine

Enabling secure communications between the client applications and MiContact Center Server requires SSL to be enabled during the installation process. Clients who have previously installed with SSL disabled and are now required to connect using SSL, must repair the Client Component pack and supply the new Enterprise address and SSL setting, by launching MiCC setup from the Start Menu or download a new copy of the CCP from CCMWeb.

To enable secure communication during the client component pack installation:

1. During installation, specify the FQDN of the MiContact Center Server or the Subject Alternate Name used in the certificate applied to the MiContact Center Server (See Figure 8).

2. Ensure that the **I want to use SSL** option is checked for new installation.
3. Follow the installation wizard to complete the installation.

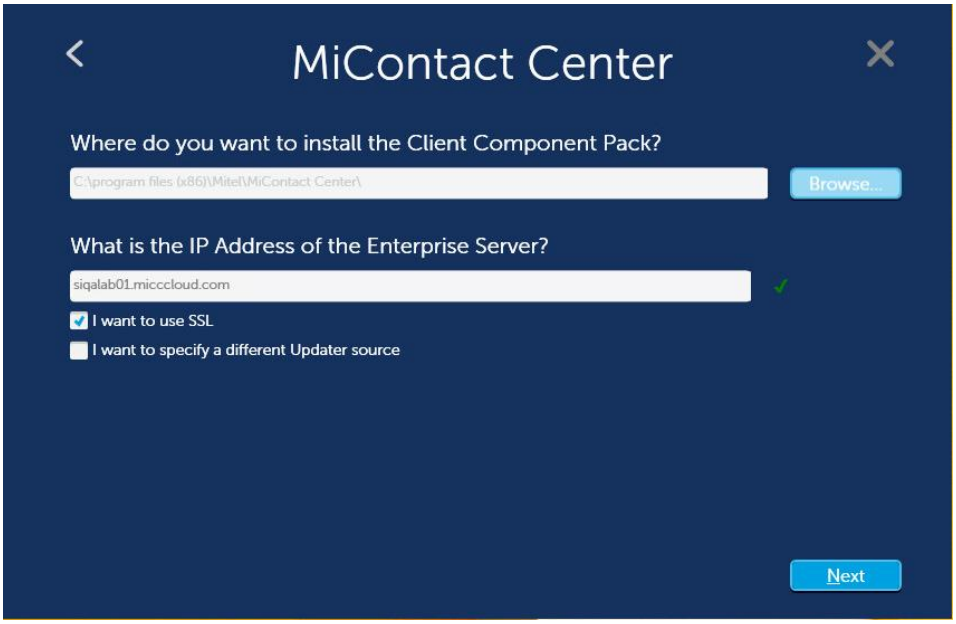


Figure 8: A sample client installation with SSL configured

To verify that the communication with the MiContact Center Server is secure, browse to CCMWeb using the shortcut placed on the desktop. You should receive no certificate errors or need to perform any additional configuration at this stage if the MiContact Center Server is secure. Clicking the Lock icon in the URL bar of Internet Explorer displays a confirmation that the certificate chain is valid (See Figure 9).

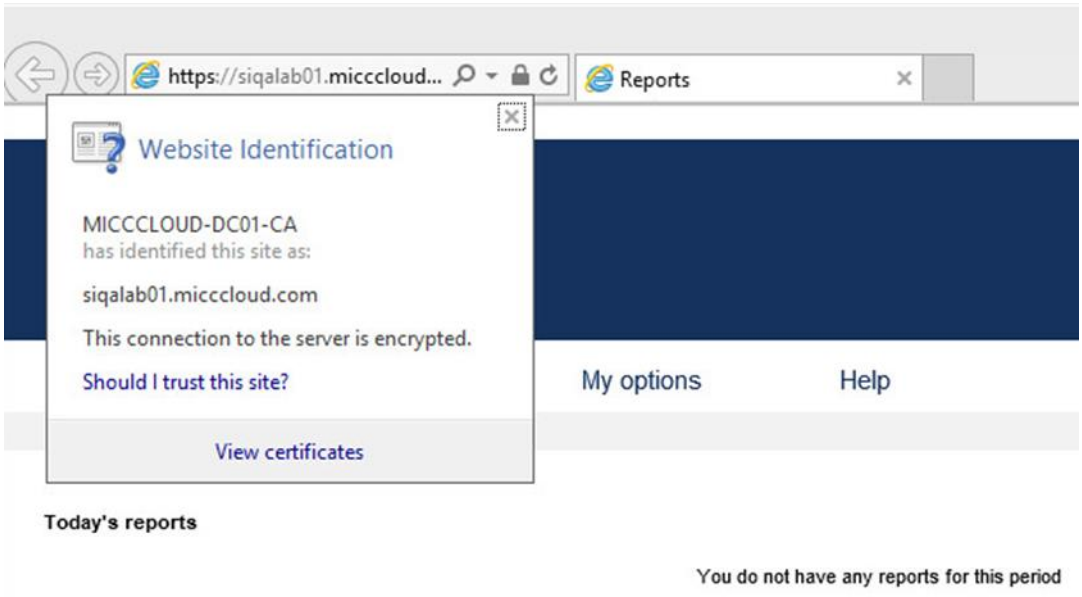


Figure 9: Site browsing to CCMWeb is secure

At this stage, you can also verify that the page connection is TLS by following these steps:

1. **Right-click** anywhere on the page.
2. Click **Properties**.

Note: In the Page Properties window, TLS 1.2 AES with 256-bit encryption (High) is displayed against Connection (See Figure 10).

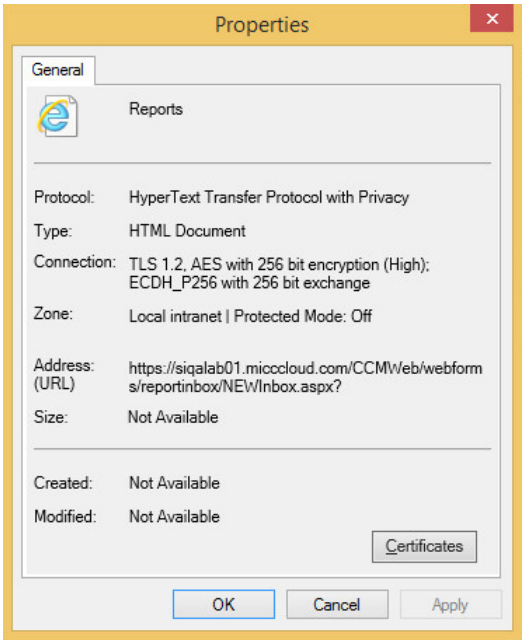


Figure 10: Page properties for CCMWeb showing TLS 1.2

Note that while communication performed with the MiContact Center Server is now configured as secure, some information still remains unsecure due to the architecture of the applications. Legacy socket connections do not communicate over web standard technologies and thus do not follow the secure communications protocols we just configured.

The unsecured communications are:

- General
  - Local configuration cache updates are transmitted in compressed binary format
- Contact Center Client
  - Real Time statistics
  - Auditor playback
  - MiTAI Commands for phone control
  - Interactive Visual Queue
- YourSite Explorer
  - Configuration changes and updates
- Flexible Reporting
- Configuration information is in compressed binary, however all other file handling for reporting is secure over CCMWeb

ALTERNATE CLIENT LOGINS

When utilizing Windows Authentication, if you want to log in as a user different from the user currently logged in to the Windows operating system, you must run the application as that user. If the process is run for the logged in user, and you specify the domain credentials of another user to log in, those credentials will be transmitted to the MiContact Center Server for verification. While this communication is done securely, if the communication channel has been compromised, the user account information will be collected.

To avoid sending login credentials to the MiContact Center Server, run the application as the user you want to log in as by following these steps:

1. Press the **Control Key** on the keyboard.
2. Right-click the application.
3. Select **Run as a different user**.
4. Enter the username and leave the password field blank.
5. Press **Enter**.

CONFIGURING SECURE REMOTE SERVER CONNECTIONS

Similarly to configuring client connectivity to the MiContact Center Server, while installing a remote server, you must specify to use HTTPS or secure HTTP when configuring the Enterprise FQDN.

To enable secure communication during the remote server pack installation:

1. Specify the FQDN of the MiContact Center Server, or the Subject Alternate Name used in the certificate applied to the MiContact Center Server (See Figure 11).
2. Ensure that the **I want to use SSL** option is checked for new installation.
3. Follow the installation wizard to complete the installation.

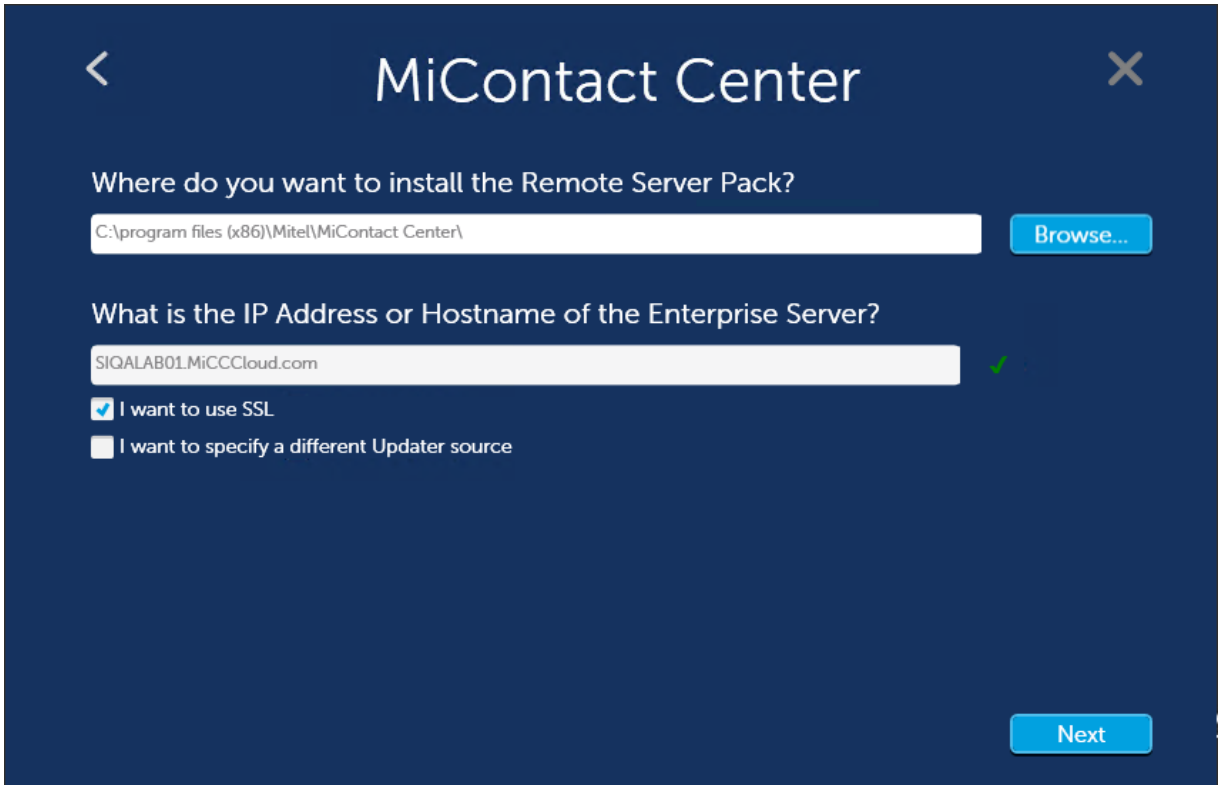


Figure 11: A remote server installation utilizing SSL

Note that while communication performed with the MiContact Center Server is now configured as secure, some information still remains unsecure due to the architecture of the applications. Legacy socket connections do not communicate over web standard technologies and thus do not follow the secure communications protocols we just configured.

The unsecured communications are:

- General
    - Local configuration cache updates are transmitted in compressed binary format
- Synchronization
    - Workflow and audio files are synchronized using HTTP REST web services or HTTPS channels

Configuration information is in compressed binary, however all other file handling for reporting is secure over CCMWeb.

CONFIGURING SECURE WEBCHAT CONNECTIONS

Using a reverse proxy enables requests to your corporate website to be served from CCMWa through your corporate website, proxying requests through the web server to CCMWa on the Enterprise Server within the corporate network. This allows implementations to obfuscate and secure connectivity from the public internet, through the DMZ, into the corporate network where chat sessions are managed.

The illustration below (Figure 12) outlines a customer initiating a chat through an external URL, which is redirected by the web server to the Reverse Proxy within the corporate DMZ infrastructure. The Reverse Proxy performs a proxy request to the MiContact Center within the corporate network. All communication is performed over HTTPS, and the certificate presented to the customer can be from Mitel.com and not the internal int.mitel.com subdomain, which allows for greater control on certificate revocation and certificate trust chain.

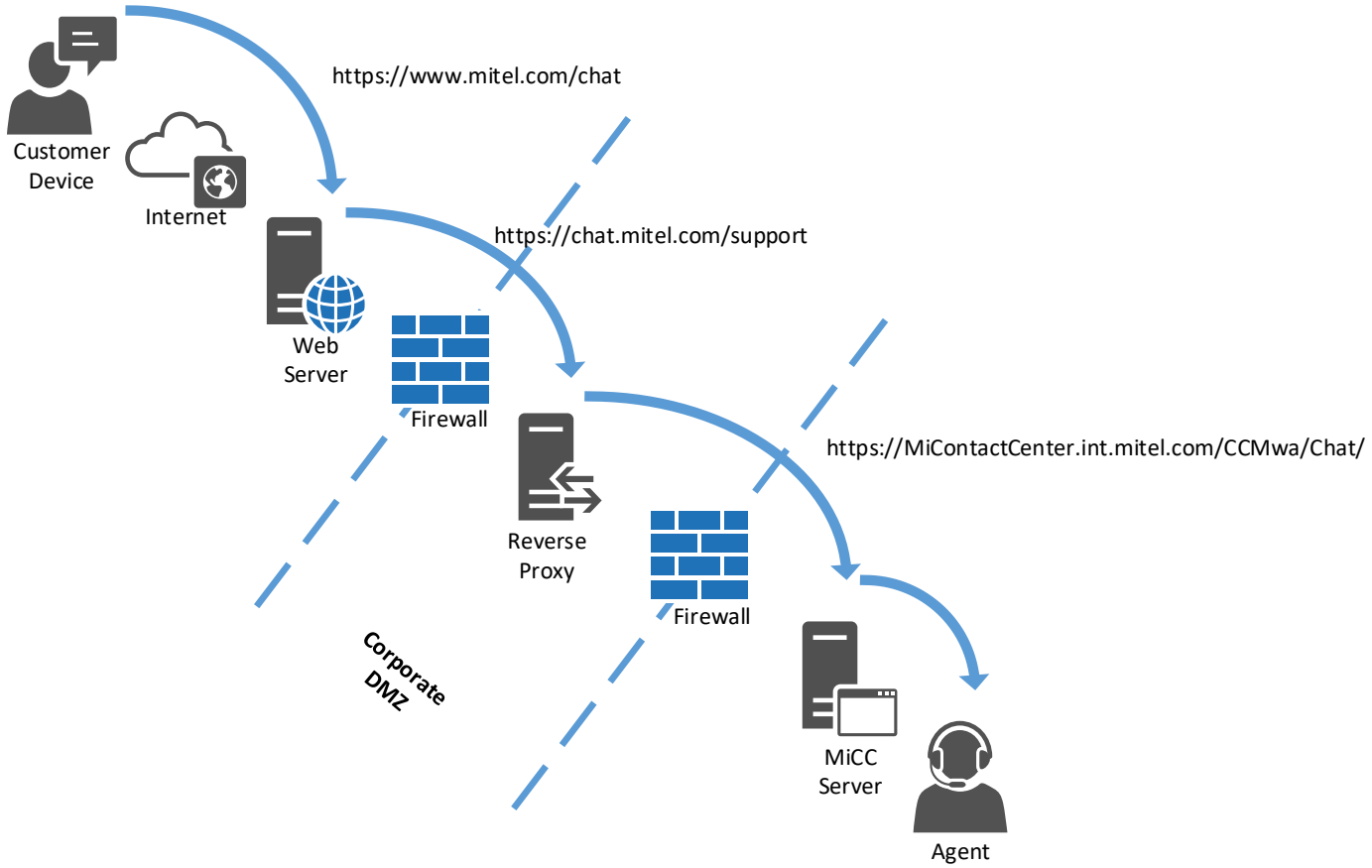


Figure 12: A sample reverse proxy with DMZ implementation

In this secure implementation, the web server and reverse proxy can be provided by any vendor, only that the MiContact Center Server must run Windows Server with IIS. This allows for a wider range of selection for the IT and Security staff for the software and hardware used to secure the network infrastructure, without impeding contact center operations and customer interactions.

For additional information on configuring reverse proxies for use with MiContact Center Chat, see the Multimedia Contact Center Installation and Deployment Guide.

CONFIGURING MICROSOFT SQL SERVER

To ensure connections to the Microsoft SQL Server are secure, Windows Authentication should be used for authenticating with the SQL Server, and client connections should be encrypted. To enable SQL Server encryption:

1. Open the **SQL Server Configuration Manager**.
2. Expand **SQL Server Network Configuration**.
3. Right-click **Protocols** for <<**INSTANCE NAME**>> where the instance name is your SQL instance.
4. Click **Properties**.
5. Change the flag for Force Encryption from **No** to **Yes** (Figure 13).
6. Click the **Certificate** tab.
7. Select the appropriate computer certificate for the Microsoft SQL Server from the drop-down list (See Figure 14).
8. Click **Apply**, and then click **OK** on the confirmation box.

9. Close the **Properties** window.

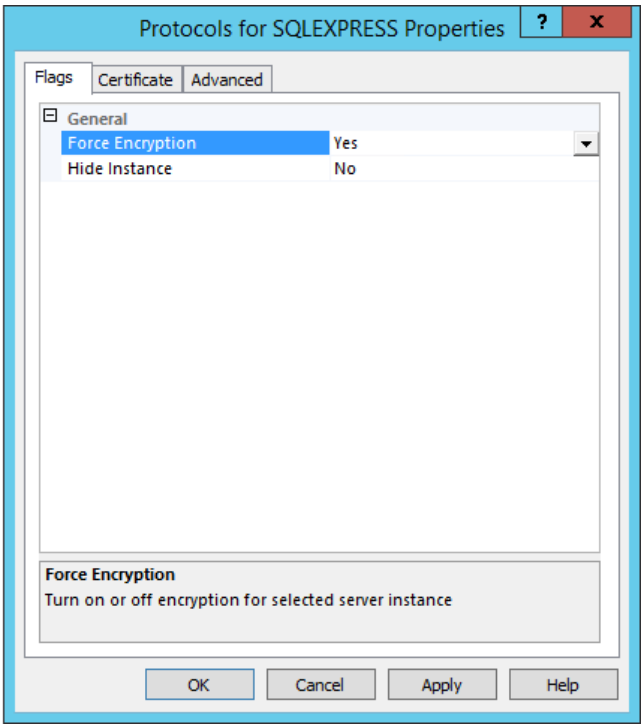


Figure 13: The Force Encryption flag set to Yes

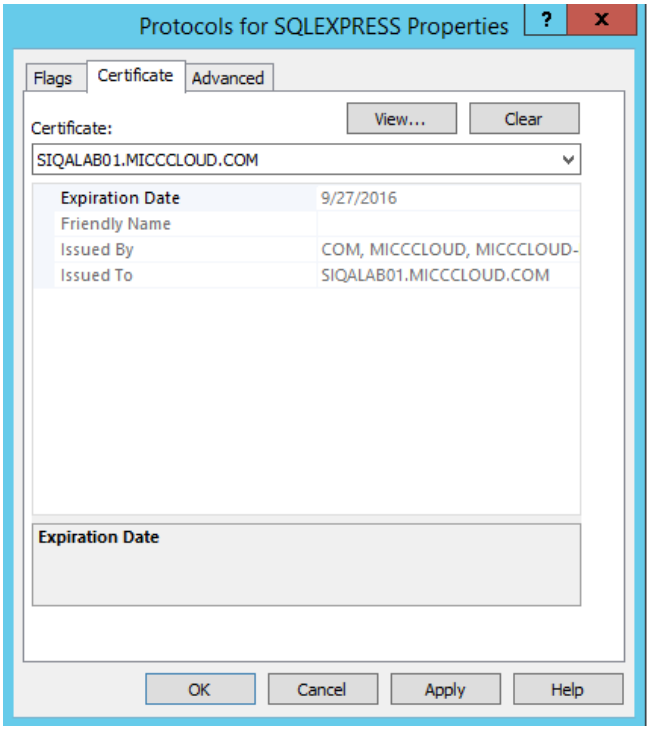


Figure 14: The server certificate applied

Now that encryption has been enabled on the SQL server, we must specify that all incoming client connections must use encryption. This ensures that clients when configured to connect without encryption, will be upgraded to a secure connection automatically. By forcing client connections to always use encryption, connection strings can remain unchanged, while connections are encrypted.

Following is the procedure to configure Microsoft SQL Server without encryption:

Note: For a 64-bit version of SQL server, all the following steps are required; for a 32-bit version of SQL server, skip steps 1 through 5 and begin at step 6.

1. Open **SQL Server Configuration Manager**.
2. Right-click **SQL Native Client** <<VERSION>> **Configuration** (62-bit) where VERSION is the version of the installed SQL Server.
3. Click **Properties**.
4. Under Force Protocol Encryption, change from **No** to **Yes** (Figure 15).
5. Click **Apply**.
6. Right-click **SQL Native Client** << VERSION>> **Configuration** (32-bit or 62-bit).
7. Click **Properties**.
8. Under Force Protocol Encryption, change from **No** to **Yes**.
9. Click **Apply**.
10. Click **OK** to close the Properties window.

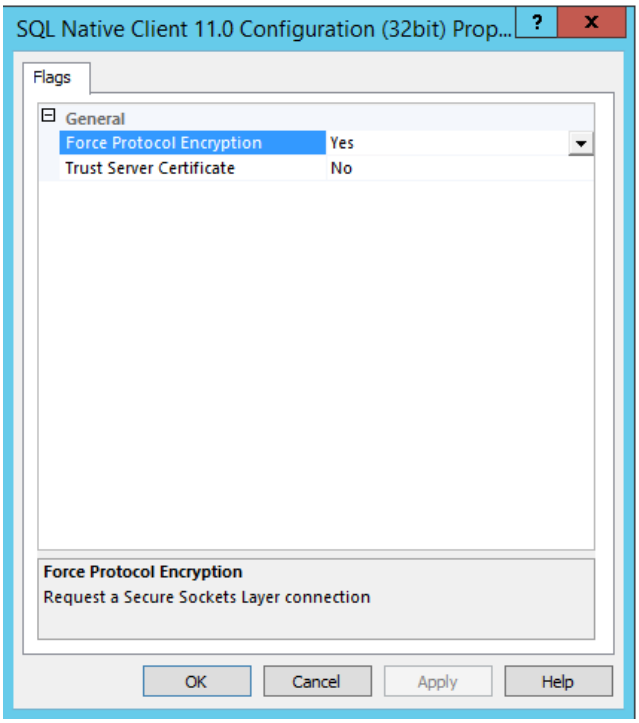


Figure 15: Native client protocol Force Protocol Encryption flag set to Yes

For these settings to take effect, you must restart the SQL Server instance. To restart the SQL Server instance:

1. In the SQL Server Configuration Manager, click **SQL Server Services**.
2. Right-click **SQL Server** << INSTANCE >> where INSTANCE is the name of the SQL Server instance.
3. Click **Restart**.

Note: If the SQL Server does not start after making these changes, it is possible that the Microsoft SQL Server service account does not have access to the private keys within the server certificate. See the article http://thesqldude.com/2011/08/03/sql-server-service-does-not-start-after-enabling-ssl-encryption/

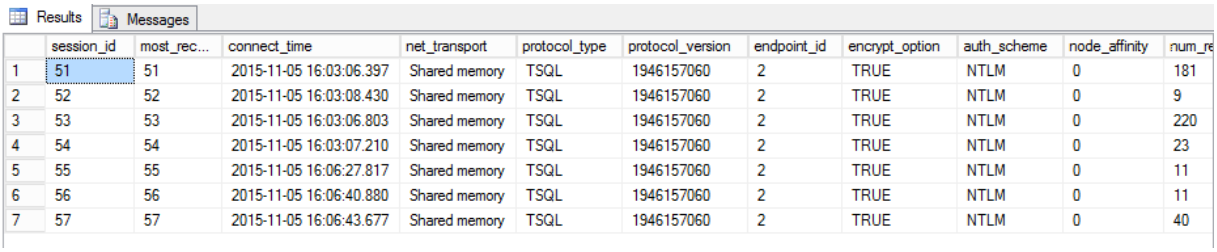
Run the following SQL command to verify that all connections to the SQL database are encrypted:

```
USE Master
SELECT *
```

FROM sys.dm_exec_connections

In the output of this command (see Figure 16):

- Connection methods are shown under the NET_TRANSPORT column.
- ENCYRPT_OPTION indicates TRUE if the connection is encrypted.
- AUTH_SCHEME indicates the authentication model used.
- NTLM indicates a Windows Authentication session.
- SQL indicates an SQL authentication session.
- CLIENT_NET_ADDRESS indicates the IP address of the connection.
- CLIENT_TCP_PORT shows the port used for the connection.



Results | Messages

	session_id	most_rec...	connect_time	net_transport	protocol_type	protocol_version	endpoint_id	encrypt_option	auth_scheme	node_affinity	num_re
1	51	51	2015-11-05 16:03:06.397	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	181
2	52	52	2015-11-05 16:03:08.430	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	9
3	53	53	2015-11-05 16:03:06.803	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	220
4	54	54	2015-11-05 16:03:07.210	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	23
5	55	55	2015-11-05 16:06:27.817	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	11
6	56	56	2015-11-05 16:06:40.880	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	11
7	57	57	2015-11-05 16:06:43.677	Shared memory	TSQL	1946157060	2	TRUE	NTLM	0	40

Figure 16: Sample query output showing secured connections

SECURING YOUR FILESYSTEM

In addition to securing connections in and out of the MiContact Center Server, it is also possible to secure the filesystem using the Anti-virus software, and/or memory, and drive encryption software such as Windows Bitlocker.

While both anti-virus and drive encryption software provide an added level of protection to your contact center server, using these can have an impact on the overall performance of the system depending on the configuration of these additional software suites.

In general, Mitel does not recommend utilizing drive or memory encryption; however, if there is such a requirement for the network deployment, you must extensively test and validate the MiContact Center to ensure that the system functions properly for all your deployment use cases.

For anti-virus software applications, it is highly recommended to add the MiContact Center installation folder as an exception to any file access scanning as scanning this folder can severely impact service and application performance. Proactive scans of the MiContact Center installation folder can be performed outside of busy hours, and/or during maintenance cycles where the server is in a low activity state.

For reference, the following is a list of the directories used in the normal operation of the contact center. Consider each of these carefully when planning your anti-virus protection strategy. If any of these directories are scanned on-demand, or on-access, MiContact Center performance may be impacted.

The MiContact Center Enterprise Server utilizes the following directories:

- <Install Drive>\ProgramData\pfdscache
- <Install Drive>\ProgramData\Mitel
- <Install Drive>\Program Files (x86)\Mitel

    - Installations upgraded from versions earlier than release 8.0 will additionally use:
      <Install Drive>\Program Files (x86)\prairieFyre Software Inc
- <Windows Drive>\Windows\System32\msmq
    - This drive is used by the MSMQ process, which is critical for performance; on-access scans should be disabled for this directory
- <Windows Drive>\Windows\System32
- <Java Install Drive>\Program Files\Java\<Java Version>\bin
    - The Java libraries are used by Elasticsearch, which is critical to MiContact Center operation
- <Windows Drive>\Windows\Microsoft.NET\Framework64\

The MiContact Center Applications utilize the following directories:

- <Install Drive>\ProgramData\pfdscache
- <Install Drive>\ProgramData\Mitel
- <Install Drive>\Program Files (x86)\Mitel
    - Installations upgraded from versions earlier than release 8.0 will additionally use:
      <Install Drive>\Program Files (x86)\prairieFyre Software Inc
- <Install Drive>\Users\<Current User>\Documents\CCMLogs
- <Windows Drive>\Windows\System32
- <Windows Drive>\Windows\Microsoft.NET\Framework64\

SUPPORT FOR SOFTWARE FIREWALLS

The MiCC Installer adds inbound firewall rules (Figure 17 and 18) for the ports required by the MiCC services and applications. Mitel recommends adding firewalls rules to the Windows firewall during installation even if the firewall is disabled.

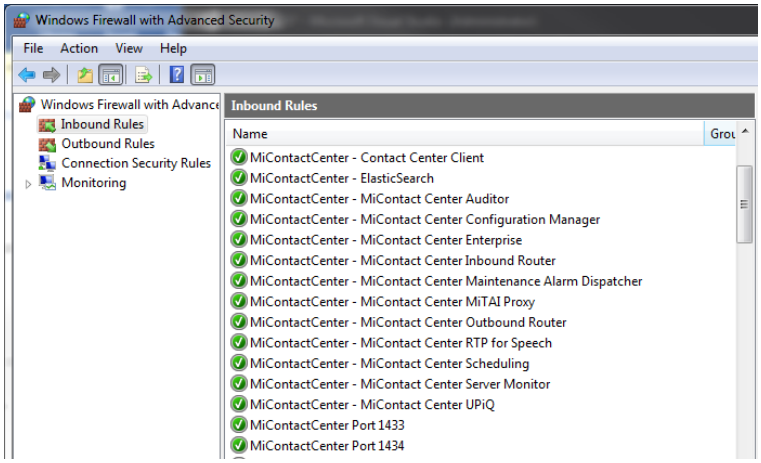


Figure 17: MITEL-Firewall

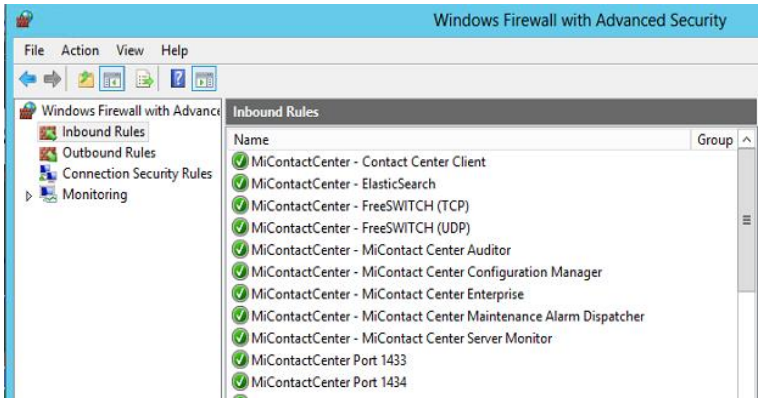


Figure 18: SIP-Firewall