Table of Contents

IBM® Secure Gateway for Bluemix™ Client

IBM® Secure Gateway for Bluemix™ client provides secure connectivity to your on-premises resources and applications.

Version: 1.7.0

Copyright and License

Trademarks

System Requirements

The Secure Gateway for Bluemix client supported in the following 64bit architecture environments.

Name	Versions
Docker	1.7.0 and greater, all supported operating systems
Docker Toolbox	1.8.0 and greater, NOT SUPPORTED

Starting with version 1.3.1 the IBM Secure Gateway client for Bluemix provides native client installation support for:

Name	Versions
Mac OS X	10.10 (Yosemite) and greater
Red Hat Linux	6.5 and greater
SuSE Linux	11.0* and greater
Ubuntu Linux	14.04 (LTS) and greater
Windows Desktop	7, 8.1, 10 and greater
Windows Server	2012 R2 and greater
Power Machine	ppc64el architecture
Ubuntu Z-Linux	-

* - SLES 11 support is only with Service Pack 4

NOTE: Only 64-bit environments are currently supported for native client installation.

Determining Hardware Requirements

The specifications of the machine running the Secure Gateway Client is largely dependent on the traffic that will be passing through each connection. Each client instance is an individual process that provides up to 250 concurrent connections. To approximate an average maximum that the machine would need to support, determine the average size of a transaction across

the client (both request and response) and then scale that to 250 concurrent transactions. Given that this number has combined both request and response sizes, the client should not exceed this memory footprint. For more precise estimates, a mixture of request sizes and response sizes should be used to better simulate a real-world transaction scenario.

For running a single instance of the Secure Gateway Client, we suggest a minimum of 2 cores and 4 GB of memory.
For HA scenarios that will be running multiple instances of the client on the same machine, we suggest a minimum of 4 cores and 8 GB of memory.

Changes

This Release

Secure Gateway 1.6.0 changes

1. ACL now supports specific path routing for HTTP/S requests
2. Destinations now support Server Name Indicators for TLS connections

Previous Releases

Secure Gateway 1.5.1 changes

1. Destination wizard added for destination creation.
2. Gateways now support uploading a custom cert/key pair.
3. Services are now limited to 50 gateways and 50 destinations per gateway.
4. Destinations/Gateways/Services can now be exported/imported.
5. Latency tests between client and server capable of being executed from Secure Gateway Server UI.

Secure Gateway 1.5.0 changes

1. Client now has a local interactive UI
2. UDP connections are now supported
3. Memory footprint of the client has been drastically reduced
4. Single client mode is no longer supported; multi will be default and transparent to a single gateway client.
5. ACL now synchronized between clients connected to the same gateway

Secure Gateway 1.4.2 changes

1. Clients are now assigned an ID when connecting to the SG server.
2. Clients can now be terminated remotely via API or SG UI.
3. The connected status of a client can be checked via API.
4. Destination-side TLS now supports Mutual Authentication.
5. Cloud destinations now support Mutual Authentication protocols.

Secure Gateway 1.4.1 changes

1. Native Windows installer is now a .exe.
2. Native Windows installer now provides options to run as a service.

Secure Gateway 1.4.0 changes

1. Multi-gateway interaction mode for the client.
2. Bi-directional connection support.
3. High availability support for the client.

Secure Gateway 1.3.2 changes

1. Native installer is provided for the Windows environment via an MSI installer.
2. All installation packages now have a companion MD5 file.
3. Upon start up the client now WARNS all users they must set the ACL before connecting any destinations.

Secure Gateway 1.3.1 changes

1. Native installer is provided for the Mac OS X environment.
2. Client interactive command changes.

Secure Gateway 1.3.0 changes

1. The client automatically starts with ACL DENY ALL. This means the user must specific an ACL file containing the access control list entries to use. Until this is done all access to on-premises resources is prevented.
2. Native installers are provided for the Linux environment.
3. Some new client commands are provided to show configuration and status.
4. Disabling and Enabling gateway and destination support.

Installation and Execution

Docker

Docker is a third-party platform that provides a container approach to installing applications quickly and easily with little or no configuration necessary. The Secure Gateway service provides a Docker image to be used after the Docker utility is installed on your workstation. To install Docker see the Docker install web site and follow the instructions for your system.

Docker Execution

To run the Docker container for IBM Secure Gateway, issue the following command:

```
docker run -it ibmcom/secure-gateway-client <gateway ID>
```

Normally it takes one parameter, your Secure Gateway gateway ID which you created in the Secure Gateway service for IBM Bluemix™.

Updating the Docker Client

Every so often you will need to update the Secure Gateway client in order to get the latest version, which can include important security updates and fixes. You are notified when an update is required.

1. To update the Secure Gateway client, issue the following docker command.

```
docker pull ibmcom/secure-gateway-client
```

2. Restart the client by issuing the Docker run command. Below is an example of just running the client without providing a gateway ID. This can be provided later using the internal interactive command line interface.

```
docker run -it ibmcom/secure-gateway-client
```

3. Note the new version update
   IBM Bluemix Secure Gateway Client version

Supported Docker Commands

The Secure Gateway Client only supports the `pull` and `run` commands for manipulating the container.

Mac OS X

Introduction

Beginning with version 1.3.1, the Secure Gateway client provides native client installation support for Mac OS X Yosemite (10.10) and above. The support is provided as the commonly used installation disk image (DMG) file. There are some prerequisites that must be installed prior to using the client on Mac OS X.

Requirements for running on Mac OS X

The following pre-requisites are required prior to running the client on Mac OS X.

1. Install Xcode development tools, it is required by some of the node modules on this platform.

Mac OS X Installation Procedure

You may require administrative privileges to perform this installation, depending on your system's security setup.

1. Mount the DMG image that was downloaded from the Secure Gateway UI, normally by 'double-clicking' it.
2. A new 'finder' window should appear. This window should contain an Applications "shortcut" icon, drag and drop the application onto the shortcut. If not, 'double-click' the mounted volume and drag and drop the application icon to the Applications icon in the Finder sidebar.

Linux

Introduction

Beginning with version 1.3.0, the Secure Gateway client provides native client installation support for selected Linux environments. A native running client will improve over-all performance in both bare metal and virtual environments. You can run the natively installed client by using the system auto-start facility or manually by using a terminal session. When you use the auto-start facility, the client automatically logs events to the defined /var/log directory and file.

The installation includes the IBM Secure Gateway client as well as a secure version of IBM's nodejs package. Both are installed under the /opt/ibm directory on the system. The installer will create or update the following:

Directory	Filename	Description
/etc/ibm	sgenvironment.conf	configuration file for upstart
/etc/ibm	passwd	creates user: secgwadmin:x:501:501::/home/secgwadmin:/bin/bash
/etc/ibm	group	creates group: secgwadmin:x:501:
/etc/init	securegateway_client.conf	upstart file
/opt/ibm	node-v<version>-linux-x64	IBM's Node JS installation
/opt/ibm	securegateway	Secure Gateway client installation
/usr/local/bin	node	symlink created for IBM's node executable
/usr/local/bin	npm	symlink created for IBM's npm executable
/var/log/securegateway	client_console.log	log file created when running client as an auto-start process

The user can run the client either manually in a terminal session or by using the built-in upstart capability.

Ubuntu Installation Procedure

You will require root or administrative privileges to perform this installation.

1. Install the Secure Gateway client. For example, if you are using a Debian package, issue the following command.

   For instance, if you are trying to install 1.4.1 version use the following command:

   ```
   sudo dpkg -i ibm-securegateway-client-1.4.1+client_amd64.deb
   ```

2. When the client installer starts, you are prompted for the following information.

   Auto-start process Yes/No

   If this is an upgrade or reinstall, you must choose whether you want the existing client to be stopped while the installation process is running. Choose Y/y to stop the existing client. Otherwise, the client package is upgraded without restarting the client, which means that you can wait for an appropriate Software Update window to perform a restart. If you choose N/n, the installation continues and you must restart the client manually.

   Gateway ID

   Set the gateway ID for the client. The gateway ID is defined when you create a Secure Gateway service. If the client fails to connect, you can change your selection by editing the .conf file.

   Security token

   If the gateway ID is enabled to check for a security token, you must provide it now. If the gateway ID does not require a security token, leave this blank.

   Logging level

   The default setting is INFO. Other valid values are TRACE, DEBUG, and ERROR.

   Access Control List

   Enter the absolute path of the file name containing the access control list on what has access to your on-premises resources. For more information, see the README.md file in /opt/ibm/securegateway or Access control list.

   Client UI

Choose if you want to use client UI. If yes, user can change the port for the UI. Default value is 9003.

Note: You do not have to answer any of the prompts, all will take the defined default or be left blank in the sgenvironment.conf file. This allows the installation process to run without user interaction.

Note: The sgenvironment.conf is read every time that you start or restart the client using the system's upstart process. You can edit the /etc/ibm/sgenvironment.conf file at any time to make changes to your configuration and restart the client to pick up those changes.

Note: The language of the Secure Gateway client service logs can be changed by changing the LANGUAGE parameter in the /etc/ibm/sgenvironment.conf file. The service logs will change to selected language after service restart.

3. If you restarted the client, to make sure that the client is running properly, issue the following command:

   cat /var/log/securegateway/client_console.log

4. If you want to edit the configuration file, to make sure that the configuration file is updated correctly, issue the following command:

   sudo vi /etc/ibm/sgenvironment.conf

5. To check the status of your client installation, issue the following command:

   sudo dpkg --status ibm-securegateway-client

An example of the output is returned:

```
Package: ibm-securegateway-client
Status: install ok installed
Priority: extra
Section: IBM/net
Maintainer: cloudoe-dev@webconf.ibm.com
Architecture: amd64
Source: ibm-securegateway+securegateway-client
Version: 1.3.0
Description: IBM Secure Gateway Client for Bluemix
IBM Secure Gateway client package for Bluemix, supports secure connections to on-premises connections.
Homepage: https://ng.bluemix.net/
License: http://www.ibm.com/software/sla/sladb.nsf/lilookup/986C7686F22D4D3585257E13004EA6CB?OpenDocument
```

Power Installation Procedure

Beginning 1.4.2, installer will be available for Power. You will require root or administrative privileges to perform this installation.

1. Install the Secure Gateway client. For instance, if you are trying to install 1.4.2 version use the following command:

   sudo dpkg -i ibm-securegateway-client-1.4.2+client_ppc64el.deb

2. When the client installer starts, you are prompted for the following information.

   Auto-start process Yes/No

   If this is an upgrade or reinstall, you must choose whether you want the existing client to be stopped while the installation process is running. Choose Y/y to stop the existing client. Otherwise, the client package is upgraded without restarting the client, which means that you can wait for an appropriate Software Update window to perform a restart. If you choose N/n, the installation continues and you must restart the client manually.

   Gateway ID

   Set the gateway ID for the client. The gateway ID is defined when you create a Secure Gateway service. If the client fails to connect, you can change your selection by editing the .conf file.

   Security token

   If the gateway ID is enabled to check for a security token, you must provide it now. If the gateway ID does not require a security token, leave this blank.

   Logging level

   The default setting is INFO. Other valid values are TRACE, DEBUG, and ERROR.

   Access Control List

   Enter the absolute path of the file name containing the access control list on what has access to your on-premises resources. For more information, see the README.md file in /opt/ibm/securegateway or Access control list.

   Client UI

   Choose if you want to use client UI. If yes, user can change the port for the UI. Default value is 9003.

   Note: You do not have to answer any of the prompts, all will take the defined default or be left blank in the sgenvironment.conf file. This allows the installation process to run without user interaction.

   Note: The sgenvironment.conf is read every time that you start or restart the client using the system's upstart process. You can edit the /etc/ibm/sgenvironment.conf file at any time to make changes to your configuration and restart the client to pick up those changes.

   Note: The language of the Secure Gateway client service logs can be changed by changing the LANGUAGE parameter in the /etc/ibm/sgenvironment.conf file. The service logs will change to selected language after service restart.

3. If you restarted the client, to make sure that the client is running properly, issue the following command:

   cat /var/log/securegateway/client_console.log

4. If you want to edit the configuration file, to make sure that the configuration file is updated correctly, issue the following command:

   sudo vi /etc/ibm/sgenvironment.conf

5. To check the status of your client installation, issue the following command:

   sudo dpkg --status ibm-securegateway-client

An example of the output is returned:

```
Package: ibm-securegateway-client
Status: install ok installed
Priority: extra
Section: IBM/net
Maintainer: cloudoe-dev@webconf.ibm.com
Architecture: ppc64el
Source: ibm-securegateway+securegateway-client
```

Version: 1.4.2
Description: IBM Secure Gateway Client for Bluemix
 IBM Secure Gateway client package for Bluemix, supports secure connections to on-premises connections.
Homepage: https://ace.ng.bluemix.net/
License: http://www.ibm.com/software/sla/sladb.nsf/lilookup/986C7686F22D4D3585257E13004EA6CB?OpenDocument

Red Hat and SuSE Installation Procedure

You will require root or administrative privileges to perform this installation.

1. Install the Secure Gateway client. For example, if you are using an RPM package, issue the following command.

   For instance, if you are trying to install 1.4.1 version use the following command:

   ```
   rpm -ivhf ibm-securegateway-client-1.4.1+client_amd64.rpm
   ```

   Note: On Red Hat Version 7 and above you will have to use the '--force' option as well.

   The client installer starts and installs the client, it will create a sgenvironment.conf file in /etc/ibm.

2. Optional: If you want to use the system's upstart process you must edit this file and provide the following for the client to start correctly. See Using Upstart for more information on editing this configuration file.

3. If you started the client using upstart, check the log file to make sure it is running correctly.

   ```
   cat /var/log/securegateway/client_console.log
   ```

4. To check the status of your client installation, issue the following command:

   ```
   rpm -q ibm-securegateway-client
   ```

Client Configuration - Editing the Auto-Start Configuration File

Edit the /etc/ibm/sgenvironment.conf file. It includes the following important variables to set:

Environment Variable	Description
RESTART_CLIENT	Stop and Restart the client during install or upgrade (Yes or No)
GATEWAY_ID	Your gateway ID as created on the Secure Gateway for Bluemix UI
SECTOKEN	Security Token for this Gateway ID, if you choose to enforce security when creating it
ACL_FILE	Access Control List File you want to use to restrict on-premises access to resources
LOGLEVEL	The log level you want to set for your service (default is INFO)
USE_UI	Set this to 'N' if you don't want to launch the client UI
UI_PORT	The port on which you want to launch client UI on (default is 9003)
LANGUAGE	The language you want to have client logs in (default is en)

Note: This file is only read if you are using your system's auto-start facility. If you are running the client manually this file is ignored.

Starting the Client

There are two ways to start the client:

- Manually. For more information, see Starting the Client Manually. Or,
- Using the system auto-start or system D facility. For more information, see Starting the client by using the system auto-start facility.

Language Support

To enable the language support for your installation, export one of the following accepted values: de, es, fr, it, ja, ko, pt-BR, zh, or zh-TW. On Ubuntu systems exporting the LANGUAGE envvar with one of these value suffices, it may be a different envvar on other supported operating systems. You can also export this envvar at the bottom of the sgenvironment.conf file in /etc/ibm if you are using the system's auto-start facility.

Starting the Client Manually

If you want to start the client manually, go to the /opt/ibm/securegateway/client directory and issue the following command.

```
   node lib/secgwclient.js <options> <gateway ID>
e.g.
   node lib/secgwclient.js --I DEBUG LIECHSVUCe1_prod_ng
```

where:
node - is the node js command
lib/secgwclient.js - the Secure Gateway client main
<options> - Command Line Parameters
<gateway id> - your gateway ID as provided from the UI

Starting the client by using the system auto-start facility

If you have chosen to use you system's auto-start facility, use one of the methods below to start the client.
Note: Auto-start functionality is not available for Red Hat Linux 6.

Additional Configuration needed for SystemV auto-start facility

System V is not setup during installation like the other auto-start facilities. Scripts are available in the installation directory /opt/ibm/securegateway/client/upstart, they are:

- secgw.sh
- securegateway_clientd

Note: This section will affect users of SuSE/SLES 11.

Optional: If you are running SuSE version 11 that uses systemV auto-start, scripts are provided that can be used to configure this process. Follow the below procedure:

- cd /opt/ibm/securegateway/client/upstart/systemV
- cp secgw.sh /usr/local/bin
- chmod +x /usr/local/bin/secgw.sh
- cp securegateway_clientd /usr/local/bin
- chmod +x /usr/local/bin/securegateway_clientd
- cd /etc/init.d
- ln -s /usr/local/bin/securegateway_clientd

- insserv securegateway_clientd
- vi /etc/ibm/sgenvironment.conf - see Client Configuration - Editing the Auto-Start Configuration File

Using Upstart

If you using the upstart capability so that the Secure Gateway client runs automatically at system startup you have to check its configuration first (/etc/ibm/sgenvironment.conf). Once you have configured the upstart service you can use the following command to start it:

```
sudo initctl start securegateway_client
```

To restart the service:

```
sudo initctl restart securegateway_client
```

Using SystemD

If you are using the systemD capability so that the Secure Gateway client runs automatically at system startup you may have to check its configuration first (/etc/ibm/sgenvironment.conf). Once you have configured the upstart service you can use the following command to start it:

```
systemctl start securegateway_client
```

To restart the service:

```
systemctl restart securegateway_client
```

For status on the service:

```
systemctl status securegateway_client
```

Using SystemV

If you are using the systemV facility so that the Secure Gateway client runs automatically at system startup see the notes at the top of this section to configure it. Once you have done this you can use YasT to start or control the daemon.

Stopping the Client

Stopping the Client Manually

If you want to stop the client manually use the 'q' or 'quit' command using the client command line interface. See the IBM Secure Gateway client CLI help for more information on stopping the client.

Stopping the client by using the system auto-start facility

If you have chosen to use you system's auto-start facility, use one of the methods below to stop the client.

Using Upstart

If you use the upstart capability so that the Secure Gateway client runs automatically, to stop the service issue:

```
sudo initctl stop securegateway_client
```

Using SystemD

If you use the systemD capability so that the Secure Gateway client runs automatically, to stop the service issue:

```
systemctl stop securegateway_client
```

Using SystemV

If you are using the systemV facility so that the Secure Gateway client runs automatically at system startup see the notes at the top of the section Additional Configuration needed for SystemV auto-start facility. Once you have done this you can use the system's systemV commands or YasT to stop or control the daemon.

Note: The logs for the Secure Gateway client running in service mode can be found at /var/log/securegateway. The logs are rolled into a new file on a daily basis.

Ubuntu Uninstall Procedure

To find the install status of the Secure Gateway client, you can use the following command:

```
sudo dpkg -l | grep ibm-securegateway-client
```

To uninstall the Secure Gateway client without removing the installation dir, configuration files, id, group id, and log files, then use the following:

```
sudo dpkg -r ibm-securegateway-client
```

To uninstall the Secure Gateway client removing everything use the following:

```
sudo dpkg --purge ibm-securegateway-client
```

Red Hat and SuSE Uninstall Procedure

To find the install status of the Secure Gateway client, you can use the following command:

```
rpm -qa | grep ibm-securegateway-client
```

To uninstall the Secure Gateway client without removing the installation dir, configuration files, id, group id, and log files, then use the following:

```
rpm -e ibm-securegateway-client
```

To uninstall the Secure Gateway client removing everything use the following:

```
yum remove ibm-securegateway-client
```

Microsoft Windows

Introduction

Beginning with version 1.3.2, the Secure Gateway client provides native client installation support for Microsoft Windows. The support is provided as an EXE installer.

Windows Installation Procedure

You may require administrative privileges to perform this installation, depending on your system's security setup.

1. Copy the EXE installer file to your system. An MD5 sum is also provided which you can use to check the integrity of the installation package.
2. Install it by either double clicking on it and answering the prompts, or using the following command in a terminal session:

   For instance, if you are running version 1.4.1 of the installer, use the following command:

   ```
   ibm-securegateway-client-1.4.1+client_windows.exe
   ```
3. The user will be prompted to choose the installation directory of their choice. The default location is the Program Files (x86) directory.
4. The user will be prompted to select the language in which they want to launch the Command Line Interface. If not language is selected, it will be defaulted to English.
5. The user will be prompted to install the client as a windows service. If the user chooses to do so, the client will run in background as a windows service with the configurations user provide in the subsequent dialog. The status of service can be checked in the service page of Control Panel. The service name is "IBM Bluemix Secure Gateway Service".
6. The installer identifies if there is an existing installation on the machine. If it is detected, the user is asked if they want to use the existing configurations, else the user has the option to enter new configurations. Details like gateway ids, security tokens, acl files and log level for each gateway can be provided here. If the user has chosen to run the client as a windows service the client will launch with the configurations provided. If the user has not chosen to launch the client as a windows service, the configurations will be stored for further use. In either case, the configurations are stored in %Installation_directory%/ibm/securegateway/client/securegw_service.config.
7. Starting version 1.5.0, the client comes with a fully functional client UI. You can configure it from the installer itself. The user can provide the password and the port number (default is 9003) from which the client UI will be launched.

   If the user chooses to launch the client with connection to multiple gateways, please take care while providing the configuration values. The gateway ids needs to be separated by spaces. The security tokens, acl files and log levels should to be delimited by --. If you don't want to provide any of these three values for a particular gateway, please use 'none'.

   Note: Please ensure that there are no residual white spaces.

   Note: Please note that the acl files should be placed in %Installation_directory%/ibm/securegateway/client directory or relative to that path.

Starting the Client as a process

To start a client, open a command window with administrator privileges.

```
cd "<Installation_directory>\ibm\securegateway\client"

secgw.cmd
```

or just double click on the secgw.cmd file.

Note: You can choose to use the configurations stored in securegw_service.config file or provide the details interactively.

Interacting with the client as a Windows service

To alter the state of the windows service, open a command window with administrator privileges.

```
cd "<Installation_directory>\ibm\securegateway\client"
```

If the service is already running, the user can use the below command to stop it

```
windowsService.cmd uninstall
```

To start the service, use the following command

```
windowsService.cmd install
```

Note: The service will run based on the configurations stored in %Installation_directory%/ibm/securegateway/client/securegw_service.config. The application logs for windows service will be stored at %Installation_directory%/ibm/securegateway/client/logs/securegw_win_service.log. The logs are rolled into a new file on a daily basis.

Windows Uninstallation procedure

There are two ways to remove Windows installation for Secure Gateway client:

1. Removing from the installation directory: Go to the installation directory for Secure gateway client and double-click on the uninstall.exe.
2. Removing from Control Panel: Remove the Secure Gateway application from the Programs section of the Control Panel.

Usage

Interacting with the Client

There are a few ways to interact with the client:

- Through terminal command line prior to startup
- After startup, using the client's interactive command line
- After startup, using the client's local UI

Multi-gateway Functionality

Previously, there were two modes of functionality: single-gateway and multi-gateway. The single-gateway mode of functionality has been replaced by a transparent multi-gateway mode, meaning the `--multi` flag is no longer necessary during startup. New gateway connections are still created as a child process of the initial client, but the newly connected client will be selected by default, making interaction nearly identical to the single-gateway mode. If multiple gateways are connected, the user can interact with them via their worker ID, available through the command line.

Terminal Command Line Arguments and Parameters

The Secure Gateway client has command line parameters and arguments.

Parameter and Arguments	Description
<gateway ID>	Connect to Bluemix by using the gateway ID that is provided
--F, --aclfile <file>	Access control List file
--g, --gateway <hostname:port>	Used to manually select a specific gateway destination (advanced use only)
--l, --loglevel <level>	Change the log level to ERROR, INFO, DEBUG or TRACE

--p, --logpath <file>	Direct logging to a specific file
--t, --sectoken <security token>	The security token to use for this gateway connection
--P, --port <port>	The port for the UI to run on. Defaults to port 9003
--w, --password <password>	The password to protect the UI with. Defaults to no password
--noUI	Prevent the UI from starting up automatically
--allow	Allows all connections to the client. Is overridden by the ACL file, if provided
--service	After an initial connection, the parent will restart within 60s if all child clients are terminated

Note: --service, --allow, and --noUI flags should be the last parameters in the command line arguments.

In order to pass these to multiple gateway connections, they must be passed in using a specific format. The gateway IDs can be passed in the same way as with the single-gateway client (with each gateway ID separated by a space); however, the order of these IDs determine the order the rest of the arguments must follow. When passing in any of the other arguments, they must be separated by -- in order to be picked up correctly. If nothing is passed in for a particular flag, it is assumed to not apply to any of the gateways.

If there aren't enough arguments supplied to fulfill all of the gateway IDs, then they will be applied in order until there are no more arguments. For example, if two gateway IDs are passed in and a single security token is passed in, then the token will be applied to the first gateway ID and not the second one.

If gateway IDs are provided that require different arguments, then the keyword none should be designated in the place of any particular argument that a gateway is not enforcing/supplying. For example. if three gateway IDs are passed in and the user would like to specify a loglevel for the first and third, the argument would look like --loglevel DEBUG--none--TRACE. In this case, none would then default to INFO.

As an example of a full command, this is how a connection command would look when attempting to connect two gateway on startup, where the first does not require a security token but the second does, where the first will be provided an ACL file and the second will not, and providing specific loglevels and logpaths to each:

```
myGatewayID_1 myGatewayID_2 -t none--<token for gateway 2> -l DEBUG--TRACE -p <full path to log file for gateway 1>--<full path to log file for gateway 2> -F <full path to ACL file for gateway 1>
```

Internal Client Interactive Command Line Arguments and Parameters

The Secure Gateway client has a command line interface (cli) and shell prompt for easy configuration and control. The interactive environment supports a richer sets of capabilities than the command line arguments, this is to facilitate better interactive control over the client.

Interactive Commands	Description
A, acl allow <hostname:port> <worker ID>	Access control list allow
D, acl deny <hostname:port> <worker ID>	Access control list deny
N, no acl <hostname:port> <worker ID>	Remove an access control list entry
S, show acl <worker ID>	Show all access control list entries
F, acl file <file> <worker ID>	Access control List file
C, displayconfig <worker ID>	Display the current Secure Gateway configuration, if available
a, authorize <worker ID>	Toggle the override of the rejectUnauthorized parameter for outbound TLS connections for the specified worker
t, sectoken <security token>	The security token to use for the next gateway connection
c, connect <gateway ID>	Connect to Bluemix by using the gateway ID that is provided
l, loglevel <level> <worker ID>	Change the log level to ERROR, INFO, DEBUG or TRACE
p, logpath <file> <worker ID>	Direct logging to a specific file
r, reverse <worker ID>	List the ports the client is currently listening on for reverse destinations
k, kill <worker ID>	Terminate the specified worker
e, select <worker ID>	Specifies a worker to perform commands on unless otherwise specified
d, deselect	Deselects the previously specified worker. Issue select command to specify another
w, password <old password> <new password>	Set the UI password. If <new password> is blank, no password will be enforced. <old password> required on password update. Passwords must contain only letters
P, port <new port>	Change the port that the UI is listening on
u, uistart <initial password> <port>	Starts the UI on localhost:<port>/dashboard. If <initial password> is blank and no other password has been set for the session, no UI password will be enforced. If <port> is blank the UI will be reachable on 9003
U, uistop	Closes the UI associated with this client session. The session will only be accessible via CLI until a new UI is manually started
R, revoke	Clears all UI authorizations associated with this client session
q, quit	Disconnect and exit
s, status <worker ID>	Print the status details of the tunnel and its connections
L, list	Displays a list of the currently associated workers

Note: If a connection has been specified with the select command and another command is called without providing a worker ID, the command will attempt to run on the connection specified by select.

Configuring your client-side TLS connection

If you want to extend TLS security all the way back to your on-premises or cloud resource such as your database, you can do so by enabling client TLS. This means that the resource would require TLS or that users can reach an HTTPS backend.

Enabling TLS security between the client and on-premises or cloud resource is separate to application-side TLS. Application-side TLS secures access between your Bluemix app and the cloud environment client. You can use client-side TLS connectivity independent of application-side TLS, or vice versa, or together to provide complete security from your Bluemix app to your resource.

Access Control List

The Secure Gateway client provides embedded ACL support. You can allow or restrict (deny) access to on-premises resources by making modifications to the ACL for the client. This can be done interactively using the client commands or specifying a file that contains the ACLs you want to have in affect.

Starting with v1.5.0, Access Control List rules will be synchronized across all clients connected to the same gateway. With this, you only need to establish/update your ACL from a single client and it will be shared across all running clients connected to that gateway. The ACL will also persist across sessions, such that connecting a new client will also apply the same ACL rules.

Note: Starting with Secure Gateway Version 1.3.0 ACL DENY ALL is in affect until the user has supplied an ACL to allow access to on-premises resources.

Access Control List commands

The supported ACL commands are:

```
acl allow [<hostname>]:[<port>]
acl deny [<hostname>]:[<port>]
no acl [<hostname>]:[<port>]
```

```
no acl
show acl
```

The forms where you have left out either a hostname or port this implies all hostnames or ports. For example, the following is an ACL rule to allow all hostnames for port 22.

```
acl allow :22
```

The following is an ACL rule to allow all hostnames for all ports, essentially disabling ACL support, this is not recommended.

```
acl allow :
```

The 'show acl' command will show the currently set ACL or provide a message on the overall setting.

HTTP/S Route Control using the ACL

Starting with v1.6.0, HTTP/S destinations can also enforce specific routes on the ACL entries. These are added the same way as the typical ACL entries, but with the path appended to the end of the rule. For example, the following will allow only requests following the /my/api path to pass through:

```
acl allow localhost:80/my/api
```

With this rule in place, requests to <cloud host>:<cloud port>/my/api* will be allowed through.

Routes are only supported on acl allow commands.

Access Control List Precedence

After providing an acl allow : command, if further acl allow commands are entered, the ALL:ALL allow rule (from acl allow :) will be removed from the list on the assumption that you no longer want to allow unbounded access. After providing an acl deny : command, if another acl deny command is entered, the ALL:ALL deny rule (from acl deny :) will be removed from the list on the assumption that you no longer want to restrict all access. If you list your current ACL rules via the show acl command in the CLI, there will be an indicator to display whether unlisted rules are being allowed or denied.

Access Control List (ACL) file

The Secure Gateway client provides embedded ACL support. You can supply a filename that contains the supported ACL commands that will be read by the client upon startup. This file should have commands of the following
format:

```
acl allow [<hostname>]:[<port>]
acl deny [<hostname>]:[<port>]
no acl [<hostname>]:[<port>]
no acl
```

Note: 'no acl' without any other parameters RESETS the ACL table and sets the access to DENY ALL.

> Please find a sample acl file in the client folder of the installation directory.

Copying your ACL file into the Secure Gateway Docker client

The Secure Gateway docker client essentially runs in it's own virtualization container. The filesystem of the hosting machine is therefore not directly accessible to processes that run inside the container, including the Secure Gateway client. Starting with version 1.8.0 of the Docker Engine, you can use the 'docker cp' command to push files that exist on your host into the container while it is running or stopped. This must be done in order to use the Secure Gateway client's ACL FILE interactive command.

To use the interactive 'cp' support in docker from your host to the docker instance you must be at docker 1.8.0. You can check this using:

```
docker --version
```

Once you have done this, your version should display as follows. It is recommended that you allow docker to run as non-root user, so run the command that is suggested after you have upgraded you engine to 1.8.0 or 1.8.2.

```
Client:
 Version:      1.8.2
 API version:  1.20
 Go version:   go1.4.2
 Git commit:   0a8c2e3
 Built:        Thu Sep 10 19:21:21 UTC 2015
 OS/Arch:      linux/amd64

Server:
 Version:      1.8.2
 API version:  1.20
 Go version:   go1.4.2
 Git commit:   0a8c2e3
 Built:        Thu Sep 10 19:21:21 UTC 2015
 OS/Arch:      linux/amd64
```

Then to push out your acl file list to the docker image follow these steps:

- Run 'docker ps' command to find your container ID

  ```
  CONTAINER ID IMAGE                          COMMAND                CREATED         STATUS  PORTS NAMES
  764aadce386b ibmcom/secure-gateway-client "node lib/secgwclient" 27 seconds ago Up 26 seconds condescending_nobel
  ```

- Copy your acl.list using the 'docker cp' command using either the container ID or name:

  ```
  docker cp 01_client.list 764aadce386b:/root/01_client.list
  ```

- Next, in the secure gateway client running in docker:

  ```
  cli F /root/01_client.list

  [2015-10-01 08:12:30.091] [INFO] The current access control list is being reset and replaced by the user provided file: /root/01_client.list
  [2015-10-01 08:12:30.093] [INFO] The ACL file process accepts acl allow 127.0.0.1:27017
  [2015-10-01 08:12:30.094] [INFO] The ACL file process accepts acl allow 127.0.0.1:22
  ```

- Display the ACL

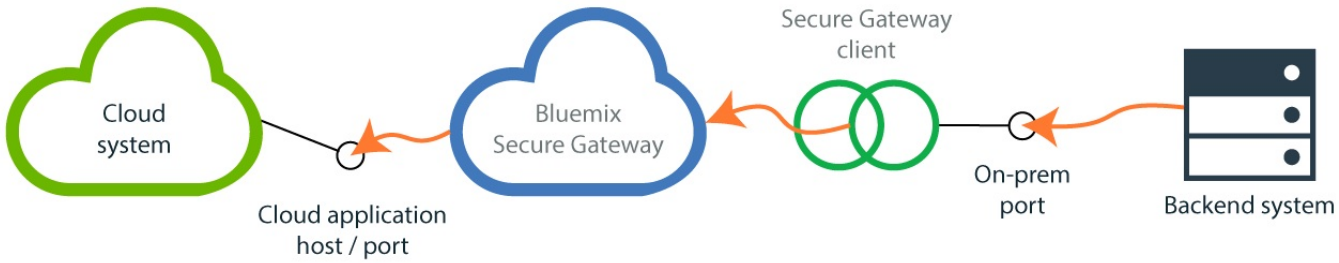
  ```
  cli> S
   ------------------------------------------------------------------
           -- Secure Gateway Client Access Control List --

   hostname                        port                value
   127.0.0.1                       27017               Allow
   127.0.0.1                       22                  Allow
   ------------------------------------------------------------------
  ```

Bidirectional Support

By combining the traditional on-premises destination with the new cloud destination, we achieve complete bidirectional support. The new cloud destination allows an on-premises service/application to send information/requests to a cloud application via the Secure Gateway Client. When creating a new cloud destination, the destination will contain resource host (the hostname or IP of the cloud application), resource port (the port that the cloud application will be listening on), and client port (the port that the Secure Gateway Client will be listening on). Once this destination is established, the on-premises service can then connect to the client port and begin sending data which will be sent through the client, to the Secure Gateway Server, and then to the specified cloud application. These destinations support the following protocols: TCP, UDP, TLS: Server Side, HTTP, and HTTPS.

The client port that is provided for a destination must be unique within the associated gateway. E.g., a single gateway cannot have two reverse destinations listening on port 12000, but two gateways can each have their own destination listening on port 12000. However, in the latter case, if both of the gateways are connected to the same multi-gateway client, only one of the destinations will be able to listen on port 12000.

Reverse Destination



On Premises Destination

High Availability

To achieve high availability, create multiple client connections to the same gateway ID. This can be accomplished by connecting multiple single-gateway clients to the same gateway ID and/or by issuing multiple connections on a single multi-gateway client to the same gateway ID. Connections will be distributed between all connected clients in a round-robin fashion.

Remote Client Termination

If a client has been provided an ID, then it can be remotely terminated via the SG UI or through the SG API. If you terminate a client that is running as a service, the client will restart and obtain a new client ID; however, if the service has multiple clients connected, the terminated client will not restart until all of the remaining clients have been terminated.

Client UI

Note: The Client UI is not supported when using Docker on Windows or MacOS.

The client UI provides a web interface for the user to interact with the Secure Gateway Client rather than the CLI. The UI is split into the following pages:

Connect

This is the initial landing page for the UI where a user can provide a gateway ID and security token to connect their first client.

Login

This page will be displayed if the UI has been password protected. If this page is reached while no password is being enforced, please refresh the page to be redirected.

Dashboard

This is the main page once a client has been connected. From here, you can access the View Logs page, the Access Control List page, and the Connection Info page. At the bottom, you can also choose to disconnect one/many of the connected clients. On the top of the page, the currently selected client will be displayed as well as an option to connect additional clients.

View Logs

This page will allow you to see the logs being generated by the selected client (shown in the upper right of the page). The displayed logs can be filtered by the check boxes below the logs.

Access Control List

This page will allow you to manipulate the Access Control List for the selected client (shown in the upper right of the page). Rules can be individually added to the allow/deny tables or a file can be uploaded at the bottom of the page.

Connection Info

This page will show the current connection information for the selected client (shown in the upper right of the page). Information such as gateway description, number of current connections, and reverse destination listeners can be seen here.

Connection Limitations

Our plans have the following concurrent limitations:

- Standard: 250 concurrent connections per client

DataPower Client Limitations

The Secure Gateway DataPower Client is in the process of being updated to match the capabilities of the rest of our clients. It currently has the following limitations:

- ACL will default to ALLOW ALL
- Does not support enabling/disabling gateways or destinations from the SG UI; however, the Administrative State option in the DataPower UI functions as an on/off switch for that particular client
- Does not support Client High Availability through the connection of multiple clients to the same gateway
- Does not support full certificate chains with destination-side TLS
- Does not support connection status polling for real-time connected and disconnected gateway status updates
- Does not support Cloud Destinations

Troubleshooting

The following section is for troubleshooting issues that you may come across during client setup or operations.

What are the best practices for running the Secure Gateway client

- Run the Secure Gateway client on an operating system (OS) partition that has network visibility of the services that are bridged by the client itself. For instance, some hosted virtualization environments support multiple network connectivity modes, including NAT and Bridged. Be sure to choose the correct connection type that provides you with access to the IBMÂ® Bluemix services from the internet.
- Install the Secure Gateway into your IT environment where your corporate security policy allows. This would typically be in a protected yellow zone or DMZ where your company can institute the appropriate security controls to protect on-premises assets. Always follow your corporate security policies and instructions when you install the Secure Gateway client.
- Before you install a client into your environment, ensure that both the internet and your on-premises assets are accessible and all host names are resolvable by a DNS. The client uses outbound port 443 and port 9000 to connect to the IBM Bluemix environment. Ensure you check or modify additional firewall and IP Table rules that might apply

Configure your Docker client to restart when your server restarts

What is happening

When you restart the server where your Secure Gateway client runs, you must manually restart the Secure Gateway Docker client. How can you get the client to start automatically after a restart of the system?

How to fix it

- On Linux or UNIX systems:
  - Integrate the Docker command into a script that can be called as a result of a CRON job.
  - If you are using a Linux work station, you can create an upstart configuration to ensure that the client is started every time that the server is restarted. For more information, see Automatically Start Containers in the Docker website.
- On Windows systems, issue the following command to start the client:

```
for /L %i in (0,0,0) do docker run -it ibmcom/secure-gateway-client <gateway_id>
```

Connection error message: Host: <host name>. is not cert's CN: <mycommonname>

What is happening

You are trying to implement on-premises client-side TLS by using the Secure Gateway client and you receive the following error message.

[ERROR] Connection #<connection ID> had error: Host: <host name>. is not cert's CN: <mycommonname>Where:

connection ID is a client assigned connection number.
host name is the host name for the connection.
mycommonname is the FDQN or common name that is used in the certificate.

Why it is happening

The Common Name, for example, the server FQDN or YOUR name, between your on-premises application and the certificate that you uploaded into Bluemix for this destination do not match.

- Check the following items:
  - You generated the certificate correctly and that you used the correct server FQDN or host name.
  - You uploaded the correct certificate into your Bluemix destination for this client.

    How to fix it

  - In the Bluemix UI, go to the Secure Gateway Dashboard.
  - Select your destination and click the Edit icon.
  - Click Upload certificate.
  - Upload the PEM certificate file that is to be used to connect to the on-premises system.

The CN in the certificate presented is the IP address of the gateway, but the certificate does not have a SAN matching the IP address and the client fails to connect

What is happening

Due to hostname resolution issues we are using the IP address in our destination. The CN in the certificate presented is the IP address of the gateway, but the certificate does not have a SAN matching the IP address and the client fails to connect
You have created a destination using TLS, but instead of using the destination's hostname you have used it's IP address. When connecting the client the following error is being thrown.

[2015-10-15 13:00:04.866] [INFO] Connection #0 is being established to 10.3.20.31:443
[2015-10-15 13:00:05.426] [ERROR] Connection #0 to destination 10.3.20.31:443 had error: IP: 10.3.20.31 is not in the cert's list:
[2015-10-15 13:00:05.427] [INFO] Connection #0 to 10.3.20.31:443 was closed

Why it is happening

What is going on is that the SSL verification code in the gateway client is treating this destination differently because it uses an IP address rather than a hostname. Instead of matching with the cert's CN, it is looking in the cert's SAN for a match of the IP address. Since there is no SAN in the cert, it sees it as a bad connection and fails the SSL handshake.

How to fix it

If you look at the error message it does not say CN, (e.g. [ERROR] Connection # had error: Host: . is not cert's CN:), but the cert's list, which leads me to believe you have generated your self-signed cert incorrectly. The problem is generating the cert using an FQDN or CN with an IP_Address. This will not work since IP addresses are only supported when using SAN.

Method for generating a certificate with an IP as the CN with openssl:

1. create an openssl config file, I copied mine from /usr/lib/ssl/openssl.cnf
2. Add an alternate_names section to the file like below:

   [alternate_names]

   IP.1 = <my application's ip>
3. In the [v3_ca] section, add this line:

   subjectAltName = @alternate_names
4. Under the CA_default section, uncomment copy_extensions (extension copying option: use with caution):

   copy_extensions = copy
5. Gen private key

   openssl genrsa -out private.key 3072
6. Gen certificate with options about organization

   openssl req -new -x509 -key private.key -sha256 -out certificate.pem -days 730 -config
7. Combine the files

   cat private.key certificate.pem > SAN.pem
8. Load the SAN.pem file into your destination as the client-TLS cert.
9. Load the SAN.pem file into your on-premises application and restart.
10. Your destination can be configured for either TCP, HTTP or HTTPS and you cloud side application should now be able to connect to your on-premises application.

If you encounter an UNABLE_TO_VERIFY_LEAF_SIGNATURE problem, check your openssl.conf file, change the following from:

keyUsage = digitalSignature, keyEncipherment

to the default:

keyUsage = nonRepudiation, digitalSignature, keyEncipherment

Connection error message: DEPTH_ZERO_SELF_SIGNED_CERT

What is happening

You are trying to implement on-premises client-side TLS by using the Secure Gateway client and you receive the following error message.

[ERROR] Connection #<connection ID> had error: DEPTH_ZERO_SELF_SIGNED_CERT Where:

connection ID is a client assigned connection number.

Why it is happening

The destination you defined is missing a client-side certificate.

How to fix it

1. In the Bluemix UI, go to the Secure Gateway Dashboard.
2. Select your destination and click the Edit icon.
3. Click Upload certificate.
4. Upload the PEM certificate file that is to be used to connect to the on-premises system.

It is not obvious how you can load the ACL using the commandline argument and parameters using Docker

What is happening

Since Docker is a container or virtualized environment, it does not have direct access to your filesystem until the container is actually launched. This prevents it from reading your host machines filesystem until it is actually launched and running.

How to fix it

This is what you can do:

- Created a Dockerfile to include the aclfile.txt

  ```
  FROM ibmcom/secure-gateway-client
  ADD aclfile.txt /tmp/aclfile.txt
  ```
- Build a new docker image

  ```
  docker build -t ads-secure-gateway-client .
  ```
- Run new docker image (need to specify -t and -i options, otherwise you will get error, file not found or ENOENT):

  ```
  docker run -t -i ads-secure-gateway-client1  --F /tmp/aclfile.txt
  ```
- Got the following output:

  ```
  [2015-09-30 16:50:32.084] [INFO] The current access control list is being reset and replaced by the user provided file: /tmp/aclfile.txt
  [2015-09-30 16:50:32.086] [INFO] The ACL file process accepts acl allow :8000
  [2015-09-30 16:50:32.087] [INFO] The ACL file process accepts acl deny local
  ```

Developing

See the following resource for information on how to use the Secure Gateway's client with the matching IBM Bluemix resources which include a Secure Gateway UI Service and SDK.

General Overview

- General Overview

Blogs and other resources

- Securing Destinations with TLS in Bluemix Secure Gateway
- Bluemix Secure Gateway - Yes I Can Get It
- Reaching the Enterprise Backend using Bluemix Secure Gateway via SDK API
- Reaching enterprise backend with Bluemix Secure Gateway via console
- REST API

Using the IBM Secure Gateway SDK for Bluemix

If you want to manage your Secure Gateway from a node.js app, you can use the JavaScript SDK. So if you work in JavaScript and want to interact with the Secure Gateway REST interface, there is no need for you to make the REST calls. The JavaScript SDK sets up the HTTP calls and puts them into easy to use functions. It is available as an NPM project here:

- IBM Secure Gateway SDK for Bluemix NPM Project

IBM Secure Gateway SDK for Bluemix Open Source

The SDK for IBM Secure Gateway is also provided as an open source project in GIT. It is available as a GIT open source project here:

- IBM Secure Gateway SDK for Bluemix GIT Project