Persistent Security

SafeGuard LM 5.6 License Manager Guide
(25-apr-2016)

The information in this document will guide you through installing and running the *SafeGuard LM* License Server.

The License Manager Applications:

sglmserver - the actual license server. This is what the client software communicates with over TCP/IP network protocol whether the client is running on the same computer, on a local are network (LAN) or over the Internet. The license server keeps track of the total number of licenses and how many licenses remain for each licensed component. It uses a license file to control which components are valid and for how long and how many licenses for each licensed component.

sglmd - this is the license daemon (Service on Windows). Although the license server can be run on its own, there are advantages of letting the license daemon run it. That is, if anything were to happen to the actual license server, the license daemon will detect it and attempt to restart the server, thus minimizing any impact of an uncontrolled termination of the license server. In addition, on Windows, if you want to run the license manager as a Windows Service, you will use this daemon.

sglmutil - the license manager utility application. It can be used to shutdown the license server, show what licenses the server has available, and for Windows, install and remove the license manager service.

Discussion:

The license server can be run in a terminal/cmd window for testing or to run it on a regular basis. On UNIX/Linux systems (i.e. Mac OS X, Linux) it can be run in the background from a terminal window by using the optional {-b} flag. On Windows, its best to run it as a Service. Since tracking activity with the license server may be desirable, when it is run either from a terminal window or as a Service, you can specify a log file where all the output from the server will be written. On Linux, this could also be /dev/null if output is not desired. The license server applications can be found in the bin folder.

Limits:

The current license server limitations are a maximum of 64 components, 64 maximum checked out components per user, and 256 concurrent users accessing the license server unless overridden on startup.

Usages:

sglmutil -c shutdown -f <license_file>

sglmutil -c restart -f <license_file>

sglmutil -c show -f <license_file>

sglmutil -c install -s <service_name> -f <license_file> -d <debug_log> -p <path_to_sglmd> -v <path_to_sglmserver> -m <max_connections> {-x <csv_log_file>}

sglmutil -c remove -s <service_name>

shutdown - shut down the license server running the specified license file.

restart - restart the license server running the specified license file.

show - show what licenses are being served by the license server specified by file.

install - install the license service on Windows. If any of the arguments contain spaces be sure to double quote the string. For Example, -s “my license manager”

remove - uninstall the license service on Windows. The service should be stopped first.

Usage: sglmd {-b} -v <license_server> -f <license_file> {-d <debug_log_file>} {-m <max_connections>} {-x <csv_log_file>}

This will use the license daemon to start the license server using the licenses in the license file and optionally redirect the output to a log file. The max_connections argument allows you to override the default of 256 separate client applications that can connect to the license server. Each connection is a thread. So depending on your system configuration the maximum value will change. The optional {-b} flag will place sglmd and sglmserver in the background. This option is only for non-Windows platforms.

The -x option allows your customers to create a CSV log file that can be used to create graphs of license usage over time, by user, component, version, whatever criteria they wish, and whatever graphing tools as well. We've use Google Sheets as an example.

Note: It is your / your customer's responsibility to ensure the system that the license manager is running on is properly configured to handle both the number of threads and the number of TCP sockets if the default maximum number of connections is increased above the default 256. Each additional connection requires one additional thread and one additional socket.

Usage: sglmserver <license_file> {debug_log_file | null} {max_connections | null} {csv_log_file}

This usage can be used to launch the license server by itself without any automatic restart capability. The max_connections argument allows you to override the default of 256 separate client applications that can connect to the license server. Each connection is a thread and a socket. So depending on your system configuration the maximum value will change. If you want to specify the max_connections argument, but not a debug_log_file, you will have to pass the string “null” as the second argument. If you want to specify the csv_log_file argument, but not the max_connections, you will have to pass the string “null” as the third argument. You should always use sglmd to control and launch sglmserver. Sglmd always makes sure the server is running.

The csv_log_file option allows your customers to create a CSV log file that can be used to create graphs of license usage over time, by user, component, version, whatever criteria they wish, and whatever graphing tools as well. We've use Google Sheets as an example.

The Server License File:

The server license file differs from the node-locked license file in that there is a SERVER line which contains the computer name, the id the license server is locked to and the TCP/IP port number the client applications communicate with the server on. Also, a license count is included for each component.

Here is an example server license file:

```
SERVER=localhost ID=001f5bf5b08f IDTYPE=ethernet PORT=29750
NAME=tree VERS=4.0 ID=001f5bf5b08f IDTYPE=ethernet COUNT=12 EXPIRES=never
CERT=e7fdb38c0e90c925c28368decf979e19
NAME=monkey VERS=4.0 ID=001f5bf5b08f IDTYPE=ethernet COUNT=12 EXPIRES=never
CERT=e6fcad8c0e95a357d5f57accc1ee9c11
NAME=banana VERS=4.0 ID=001f5bf5b08f IDTYPE=ethernet COUNT=12 EXPIRES=never
CERT=a287a6e27791de5cb88077a9d4eb87
```

The server license file locks this entire license file to the computer with this ethernet address. In this example, three components are being served “tree”, “banana” and “monkey”. There are twelve total licenses for each of of the three components that can be checked out at one time. If more than twelve licenses are tried to be checked out, the thirteenth will fail.

Once the license file is generated for the desired computer, start the license manager using that file.

Here is an example of what it might look like:

```
$ sglmd -v sglmserver -f float.lic
  2016/01/07 07:35:57 Server starting on localhost, port 29750, max components 64, max connections 256

 Action    Component      Version   Count   Expires
 Enabling  tree           4.0       10      never
 Enabling  monkey         4.0       10      never
 Enabling  banana         4.0       10      never
 Ready...
```

Serving different versions of the same component can be done by one license server. The important thing to remember here is that you want to list the duplicate component line versions numbers in numerical order from lowest to highest. That way, when a version 4.0 application attempts to check out a license, it will obtain the VERS=4.0 components first before spilling over to the 5.0 licenses. If you had them in reverse order, the VERS=5.0 license would be checked out first by version 4.0 of the software.

When version 5.0 of the software requests a 5.0 license, then only the 5.0 licenses are consumed.
Here is an example of a license file with duplicate components and different versions.

```
SERVER=localhost ID=001f5bf5b09f IDTYPE=ethernet PORT=29750
NAME=monkey VERS=4.0 ID=001f5bf5b09f IDTYPE=ethernet COUNT=12 EXPIRES=never
CERT=c6bf634c55dff68e3417eeaa6abd36dfcb
NAME=monkey VERS=5.0 ID=001f5bf5b09f IDTYPE=ethernet COUNT=12 EXPIRES=never
CERT=c0d918435bbfee94227888c772c948d8d3
```

Expired licenses

If a license expires over time when the license manager is running, the particular license(s) will be disabled. The license server will check once ever hour for this. So depending on what minute the license server is started, once an hour, it will scan the existing licenses to see whether they have expired. If they have expired, the server will log a message "DISABLING" of the license in question. It will also display the exact reason for the disabling of the license in case it was for a different reason.

How does a client application find the License Server?

The client uses one of two formats. The first is in the format PORT@HOST. Where port is the port number the client application tries to connect to the license server on, and host is the hostname where the license server is running. The default port number for *SafeGuard LM* is 29750. The default hostname is "localhost". So, 29750@localhost is the default *SafeGuard LM* license path.

The second way for the client application to find the license server is to just specify the actual license file.

In either case, the user can specify an environment variable SG_LICENSE_FILE that contains either the port@host or the actual license file path. This environment variable overrides the internal default.

As a developer you should specify the default license path in your application using sgSetAttrStr(SG_ATTR_LICENSE_FILE,....) and specify either the port@host or the license file that is placed in your installation folder somewhere. The end-user would still be able to override it with the environment variable unless you disable the override by calling sgSetAttrInt(SG_DISABLE_ENV,1) which will not allow the user to change the location. In practice, this is not a good idea.

License Server Defaults:

The default TCP/IP timeout for license connections is 2 seconds. This can be overridden by using sgSetAttrInt(SG_ATTR_TCP_TIMEOUT,N); where N is the number of seconds.

The default maximum acceptable time difference between the client and license server is 36 hours. This can be overridden by using sgSetAttrInt(SG_ATTR_MAXIMUM_TIMEDIFF,N); Where N is the number of hours. Valid hours are 1 to 336 (two weeks).

The default retry interval for lost connections to the license server is 2 minutes. This can be overridden by using sgSetAttrInt(SG_ATTR_RETRY_INTERVAL,N); Where N is the number of minutes between reconnect attempts.

The default retry count is 5. This is the number of times the client will attempt to reconnect to the license server. This can be overridden by using sgSetAttrInt(SG_ATTR_RETRY_COUNT,N); Where N is the retry count.

Note: licenses are checked out on a per process/connection. So a single process "A" requesting license "foo" will checkout one "foo" license each time sgCheckout() is called. SgCheckin() will check in one license each time it is called until the count reaches zero for that process.

Proxy Server

SafeGuard LM includes a multi-threaded reverse proxy server. It can be used to forward TCP/IP requests from the Internet to the floating license manager and the Product Activation Server that might not be directly accessible from the Internet. A reverse proxy takes requests from the Internet and forwards them to a server on an internal network.

Usage: sgproxyserver <local_port> <remote_host> <remote_port> <max_connections>