Persistent Security

SafeGuard LM 5.6 Product Activation Server
(25-apr-2016)

The information in this document will guide you through installing and running the *SafeGuard LM* Product Activation Server (PAS). The PAS is used to to activate your products over the Internet. Wether they are demo node-locked, demo floating, or purchased licenses. You control all aspects of the activation from your site. The PAS uses MySQL to control the license distribution rules for your applications as well as storing information regarding already issued licenses. The PAS runs on a computer behind your firewall.

The PAS is accessed through an API available in the client software which can be accessed from your installation program or your application itself. If you prefer, you can access the API from multiple languages such as Java, Python or C/C++ and integrate product activation from your website.

The Product Activation Server Applications:

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

sgpautil - the product activation server utility application. It can be used to shutdown the product activation server and for Windows, install and remove the product activation service.

Discussion:

The product activation 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, Ubuntu) it can be run in the background from a terminal window. On Windows, its best to run it as a Service. Since tracking activity with the product activation 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 product activation server applications can be found in the bin folder.

Limits:

The product activation server is multi threaded and has a maximum of 32 simultaneous connections. Since connections are established for a very short time (around a second), the need for more than 32 threads is not warranted.

Usages:

sgpautil -c shutdown -s <pa_server> -t <port> -p <password>

sgpautil -c restart -s <pa_server> -t <port> -p <password>

sgpautil -c install -s <service_name> -f <control_file> -d <debug_log> -p <path_to_sgpad> -v <path_to_sgpaserver>

sgpautil -c remove -s <service_name>

sgpautil -c status -s <pa_server> -t <port>

sgpautil -c showlog -s <pa_server> -t <port> -p <password>

shutdown - shut down the product activation server.

restart - restart the product activation server.

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

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

status - displays the status of the Activation Server. Either “Running” or “Down”

showlog - displays the Product Activation Server logfile. The server must have been started with a log file.

Usage: sgpad {-b} -v <pa_server> -f <control_file> {-d <debug_log_file>}

This will use the product activation daemon to start the product activation server using the control file and optionally redirect the output to a log file. The optional **{-b}** is for non-Windows platforms and this will run the server as a daemon in the background.

Usage: sgpaserver <control_file> {debug_log_file}

This usage can be used to launch the product activation server by itself without any automatic restart capability.

The Server Control File:

The Server Control File specifies database names, table names and other items associated with database connections as well as server properties.

Here is an example server control file:

```
#
# Control file for the Product Activation Server
#

SERVER=server1 PORT=29775
# the hash method for generating licenses can be either rc4 or sha256
CERTIFICATE_HASH_METHOD=rc4
SERVER_PASS=servpass
MYSQL_HOST=127.0.0.1
MYSQL_PORT=3306
MYSQL_USER=pasadmin
```

MYSQL_PASS=secret
MYSQL_DBNAME=sgpadb
MYSQL_DEMO_NODE_TABLE=demo_node
MYSQL_DEMO_NODE_DATA_TABLE=demo_node_data
MYSQL_DEMO_FLOAT_TABLE=demo_float
MYSQL_DEMO_FLOAT_DATA_TABLE=demo_float_data

Here is an example of launching the Product Activation Server:

```
$ sgpad -v sgpaserver -f control.txt
 2014/3/1 16:01:39 Server starting on server1 port 29775
 Ready...
```

How does a client application find the Product Activation Server?

The server address and port number are specified in the API that requests the license.

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>