

Gnu-gk - The Open Gatekeeper

There are a few open-source gatekeeper implementations available on the Internet. This one was selected as it appears to be the most mature, and also because it has been tested at a limited number of Universities in the UK (and overseas) and has been found to inter-work with the GDS. The source code and the binary executables for a number of operating systems are available at:

<http://www.gnugk.org> [1]

The following refers to the installation and configuration of a gnu-gk binary file on a Windows® 2000 operating system (the equivalent to a setup.exe file). It is possible to compile and then install the source code for Windows® operating systems, or other operating systems, but for simplicity it was felt that a download of the ready-compiled binaries would suffice. The same format is used for the configuration file whatever the operating system (i.e. configuration files can be migrated between different operating systems).

The Windows® 2000 host running the gatekeeper had previously had VNC (Virtual Network Computing) remote access software, ZoneAlarm® (personal firewall), and Sophos (Anti-virus) software all loaded. All available Windows® security updates had also been installed.

Installation

To download the zipped executable file for version 2.0.7 visit

<http://www.gnugk.org/h323download.html> [2]

and select 'Windows Executable'. Select, download and save this file to a folder on the C: drive, then unzip it. The extracted files are all located in a directory called 'openh323gk'. This folder contains the following folders:

bin – this contains the executable files which run the program. It also contains the file gatekeeper.ini which configures the gatekeeper and is read each time the configuration is reloaded.

docs – this folder contains the documentation for the gnu-gk. It includes a manual and release notes etc. This is very useful although the manual on the gnu-gk website is slightly more up to date. This folder also includes a sub-folder, 'old', which contains more useful documentation, including the file 'reference' which gives a full command reference. When these files are opened in 'Notepad' they are not formatted very well. It was found to be much easier to use Microsoft® WordPad to read and print off these documents.

etc – this folder contains a number of example configuration files (again, they are clearer when opened in Microsoft® WordPad, or the DOS editor ‘Edit’).

Starting the Gnu-gk

To test that the installation has been successful, copy the following lines into Notepad and save as **gatekeeper.ini** in the [drive]:\openh323gk\bin folder (where the gnugk.exe file is also located). Note the mis-spelling of the word ‘fourtytwo’.

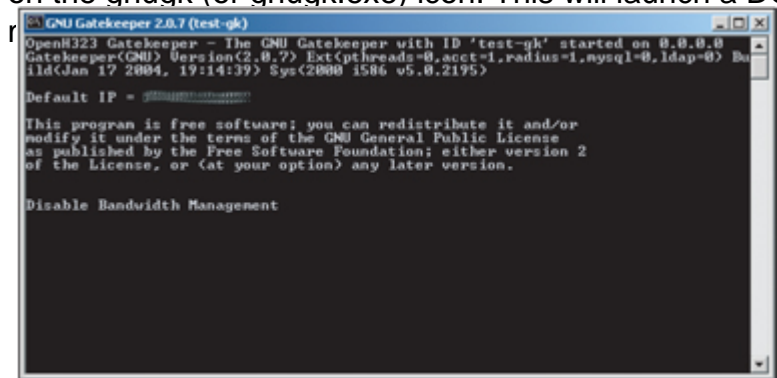
```
[Gatekeeper::Main]
```

```
Fourtytwo=42
```

```
[GkStatus::Auth]
```

```
rule=allow
```

Start the gatekeeper application by browsing to the openh323\bin folder and double-clicking on the gnugk (or gnugk.exe) icon. This will launch a DOS ‘Command’ window with a status



v active.

Figure 1. The gnu-gk start-up window

If you wish to see feedback from the gatekeeper, an alternative method of starting the gnu-gk is to use the Windows® Start/Run pop-up and browse to the gnu-gk.exe file. Something like **C:\openh323gk\bin\gnugk.exe** will appear in the **Open:** box. Add **-t** for basic reporting of the gatekeeper’s actions. Increasing the number of **t**’s will increase the verbosity of the reporting. For testing during the production of this report **C:\openh323gk\bin\gnugk.exe -tt** was generally used.

Gnu-gk Configuration

The entire Configuration of the gnu-gk is done using the **gatekeeper.ini** file. The file is divided into various sections, each covering a different area of the gatekeeper’s functionality. Each section consists of a series of statements, such as the two mentioned above. The first of these is:

```
[Gatekeeper::Main]
```

Fourtytwo=42

This is the minimal **gatekeeper.ini** file. The application looks for a configuration file and this statement confirms that it is there.

Default values

For every potential function of the gatekeeper there is a default state. When a section is invoked using square brackets [Like::This] then it is possible to choose between explicitly making statements to set the state of each parameter for that section, or only making statements which alter the default states for that section. For example, the second part of the minimal **gatekeeper.ini** file is:

```
[GkStatus::Auth]
```

```
rule=allow
```

The **[GkStatus::Auth]** section of the .ini file deals with remote access to the gatekeeper, from a different machine. The default is rule=forbid, which denies remote access to the gatekeeper from any other machine. So in this case it is necessary to explicitly set a different value which allows access from any remote machine. This is a temporary state and the final version of the .ini file will specify those machines that may have remote access to the gatekeeper (this is covered in !!REFERENCE section 8.3.5 *below #gatekeeperstatus*).

The configuration file, therefore, is built up as a series of sections, each with one or more statements. It is also possible to add comments to the file, by preceding each line of comment with a hash (#). The rest of this chapter considers each section that is relevant to a simple Janet Videoconferencing-IP scenario, in order to build up a complete **gatekeeper.ini** file for a machine deployed at a JANET-connected organisation. It is not necessary to mention every possible parameter that has been left with its default value, although important and common ones will be mentioned in the relevant sections. For a full list of possible sections and parameters/values, please see the gnu-gk manual:

<http://www.gnugk.org/h323manual.html> ^[4]

As with the other gatekeepers discussed in this report, only the functions necessary for a simple Janet Videoconferencing set-up are mentioned here. The gnu-gk also has a number of additional features that are not included here. These include: NAT (Network Address Translation) traversal, an H.323 proxy security measure, logging and producing reports.

Basic parameters

These are specified in the **[Gatekeeper::Main]** section.

Fourtytwo=42 must always be present. This confirms the presence of the **gatekeeper.ini** file. It is possible to define a name for the gatekeeper at this point. This may be the same name as the machine that the gatekeeper is running on, or some other useful name. The default name for every installation of gnu-gk is *OpenH323GK*.

Name=any-gk

By default the gatekeeper will listen out for requests on every network interface it has. If a machine has multiple network interfaces and you only wish it to listen on one of these, specify the IP address of the interface as the gatekeeper's 'home', using the statement:

Home=[xxx.xxx.xx.x]

Terminal authentication method

The **[Gatekeeper::Auth]** section is used to define the authentication method for allowing registration with the gatekeeper. As with the other gatekeepers tested, the simplest method is to authenticate on a per-terminal basis. In this case, the combination of the extension number of the endpoint and its unique IP address is used to authenticate each terminal. If these are not passed to the gatekeeper correctly in the terminal's RRQ (registration request), then the terminal will be refused registration. Specify that you are using these parameters to test for authentication, as follows:

AliasAuth=required;RRQ

default=allow

Terminal specification

The **[RasSrv::RRQAuth]** section specifies the precise E.164 'alias' and IP addresses of the terminals you wish to allow.

Gatekeepers and H.323 terminals use the RAS (Registration, Admission, and Status) Protocol to exchange messages to perform management functions. The RRQ is part of the RAS protocol and contains a number of aliases and identifiers. On receipt of an RRQ the gatekeeper will check the registration and admission parameters with which it is currently configured, and make a decision on registration based on those parameters.

The **[RasSrv::RRQAuth]** section includes a simple statement for each terminal that you wish to allow to register:

100=sigip:192.168.12.6:1720

101=sigip:192.168.12.7:1720

102=sigip:192.168.12.8:1720

If the IP address of the machine that an RRQ address is received from matches one in the above list, and is issued on port 1720 (the default port for such signals), and the E.164 extension for that signal's IP address is as stated here, then the registration will be allowed. Here, three machines have been specified by this E.164 extension/IP address combination. Add more or fewer lines with appropriate E.164/IP pairs to your file depending on the number of terminals that you wish to allow to register.

Lastly, it is necessary to state the default behaviour: what to do if the E.164/IP pair in the RRQ

does not agree with one of those listed above. In this case the registration should be rejected.

default=reject

Gatekeeper status monitoring #gatekeeperstatus

The gnu-gk outputs status messages and its responses to messages, etc. to a default external port: 7000. To monitor this output it is necessary to enter the following on a command (DOS) window on another machine (the DOS command prompt is found at Start/Programs/Accessories/Command Prompt): **telnet ggg.kkk.iii.ppp 7000** (substitute the IP address of the gnu-gk). The default state for the gatekeeper is to deny any access to this port from any machine, but the simple file at the beginning of this chapter which can be saved as a temporary measure while the gatekeeper is being configured, included the section:

```
[GkStatus::Auth]
```

```
rule=allow
```

This has the effect of letting the gatekeeper be monitored from any remote machine. Obviously, this is not a secure situation, so in this section it is possible to define those machines that you are likely to use to access the gatekeeper remotely. There are various rules that can be defined here, but the simplest is to define each IP address explicitly. Replace the **rule=allow** line with this one:

rule=explicit

Then list the IP addresses that you wish to use, with =1 appended, with no spaces:

```
192.168.23.41=1
```

```
192.168.23.42=1
```

If you wish to access this status port from the same machine that is running the gatekeeper then the local IP address will need to be included in this list. It is possible to specify a Class C IP network, but this is slightly more complex — and slightly more insecure — and so is not covered here.

Neighbour gatekeeper specification

In order for the gnu-gk to participate in the GDS (see [The Global Dialling Scheme](#) ^[5] above), it needs to know about the national directory gatekeepers maintained by the Janet Videoconferencing. For calls between terminals within the gatekeeper's zone, only the three-digit E.164 local extension number needs to be used. The gatekeeper will be able to resolve the number from the machines that it currently has registered with it. If the gatekeeper is sent an E.164 number that can't be resolved locally then it needs to be told where to send it, ie. to the national directory gatekeeper. The section of the .ini file that covers this configuration is **[RasSrv::Neighbors]**. Please note the American English spelling of the word 'neighbors'. Two lines need to be entered (using the details specified by the Janet Videoconferencing-MC for the national directory gatekeepers). These take the form:

gatekeeper-name.ja.net=aaa.bbb.ccc.ddd:1719;*

Each of these lines describes the name of the neighbouring gatekeeper, its IP address and the port on which it should be contacted. The asterisk signifies that any number that cannot be resolved locally should be sent to this neighbour. Entering two will ensure that if no answer is received from the first then an attempt will be made to access the second.

Global Dialling Scheme – location requests

The **[RasSrv::LRQFeatures]** section details the parameters for responses to LRQs from one of the neighbouring gatekeepers specified in the previous section (the national directory gatekeepers). It also specifies the parameters for forwarding LRQs to one of those gatekeepers.

The first two parameters are necessary for the successful resolution of E.164 addresses in the Janet Videoconferencing environment. The first of these forces the gatekeeper to forward a location request even if there is no 'hopCount' in the location request that it receives.

AlwaysForwardLRQ=1

The next command is turned on by default, but it should be turned off (i.e. make the value 0), in order to ensure inter-operability with a range of different gatekeepers and endpoints.

IncludeDestinationInfoInLCF=0 The default time that the gatekeeper will wait to receive a response from the neighbour gatekeeper before giving up is two seconds. This is a reasonable time, so can be left unchanged.

There is no default value for the number of times that an LRQ should be forwarded by directory service gatekeepers in order to resolve an address, so this must be explicitly stated. Nine or ten is a reasonable number so this section must have a statement:

ForwardHopCount=9

The national directory gatekeeper is a Cisco® gatekeeper, and there are subtle differences between this gatekeeper and the gnu-gk in the way the H.323 standard has been implemented. In order to overcome these, the following statement should be included:

CiscoGKCompatible=1

Global Dialling Scheme – local zone

There is no command to 'tell' the gnu-gk the E.164 address of its own local zone(s), but incoming LRQs detail the complete E.164 address of local terminals, i.e. they will include 0044xxxxx before the number that refers to the local terminal. For this reason it is necessary to configure the gnu-gk to strip off the 0044xxxxx element if it matches the zone address that has been supplied by Janet Videoconferencing. In this case the local zone number is OXXYY, so the configuration section **[RasSrv::RewriteE164]** must be invoked. This section allows for address re-writes, and usually will be of the form xxxxx=yyyyy – i.e. replace xxxxx with yyyyy. As we want the 0044OXXYY element of the address to be stripped off, but replaced with

nothing (to leave only the terminal extension), the configuration statement takes the form:

00440XXYY=

i.e. take off the 00440XXYY and replace it with nothing at all.

This statement completes the configuration file (which is reproduced in its entirety in Appendix B). This file should now be saved with all these changes and statements intact, and should be written over any existing **gatekeeper.ini** file. The gatekeeper can be stopped by entering CTRL-C in the DOS window that it opens on starting. Restarting the gatekeeper will make it re-sample the gatekeeper.ini file that has been saved and the gnu-gk will now be running with the desired configuration.

Once you are sure it is working, it is a good idea to back up the gatekeeper.ini file to another location or backup disk.

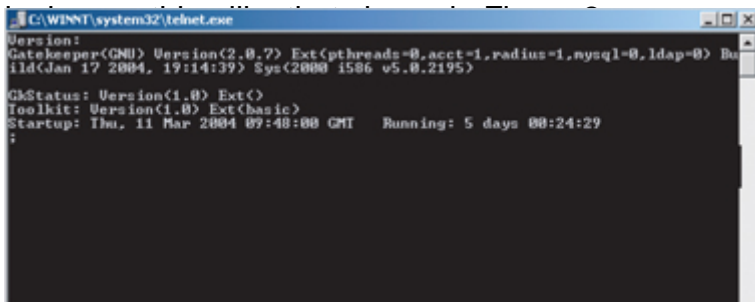
Monitoring the gatekeeper

It is possible to monitor the status of the gnu-gk by starting a telnet session from one of the machines that is specified in the **[GkStatus::Auth]** section of the gatekeeper.ini file (see !!REFERENCE section 8.3.5 above **#gatekeeperstatus**).

The command to do this is:

telnet xxx.xxx.xx.xx 7000

where xxx.xxx.xx.xx is the IP address of the machine running gnu-gk. The telnet session will

A screenshot of a Windows telnet client window titled 'C:\WINNT\system32\telnet.exe'. The window shows the output of a telnet session to a gatekeeper. The text displayed is: 'Version: Gatekeeper(GNU) Version(2.0.7) Ext(Cpthreads=0,acct=1,radius=1,nysql=0,ldap=0) Build(Jan 17 2004, 19:14:39) Sys(2000 1586 v5.0.2195) GkStatus: Version(1.0) Ext() Toolkit: Version(1.0) Ext(basic) Startup: Thu, 11 Mar 2004 09:48:00 GMT Running: 5 days 08:24:29'. The rest of the window is blacked out.

[6]

It is possible that status messages and reports will appear on this telnet session. In some cases these appear alarming, but can be ignored. For example, when a Polycom iPower™ system is connected to the gnu-gk it sends a Lightweight Registration Request to the gatekeeper as a 'keep alive' message for its registration, once every minute. However this is misinterpreted by the gnu-gk which responds with an LRJ (location request reject) message. So an LRJ appears on the status line in this telnet session once every minute, despite the fact that the iPower™ has registered and is operating normally. A full list of the status messages that you might see, and what they mean, is included in the gnu-gk manual.

A number of useful commands can be sent to the gatekeeper using the telnet command line. A complete set of the available commands appears in the gnu-gk manual, and entering 'h' will also display the full list. Some of the most useful are shown in the table *below*.

Table 1: Useful gnu-gk telnet status line commands

Command	Abbreviation	Meaning
Reload		Reload the configuration (the gatekeeper.ini file)
Version	v	Show gatekeeper and
Statistics	s	Show current operating
Printallregistrations	r	Show currently registered
Find <alias>	f <alias>	Find a particular term (for example, its E.164)
Exit	q	Quit the status port

Gnu-gk references

- gnu-gk Home Page: <http://www.gnugk.org/> [1]
- gnu-gk manual: <http://www.gnugk.org/h323manual.html> [4]

Source URL: <https://community-stg.jisc.ac.uk/library/janet-services-documentation/gnu-gk-open-gatekeeper>

Links

[1] <http://www.gnugk.org/>

[2] <http://www.gnugk.org/h323download.html>

[3] <http://community.ja.net/system/files/gkconfig01.jpg>

[4] <http://www.gnugk.org/h323manual.html>

[5] <https://community.ja.net/library/videoconferencing-booking-service/global-dialing-scheme-explained>

[6] <http://community.ja.net/system/files/gkconfig02.jpg>