NightStar RT Installation Guide

Version 3.2

(RedHawk Linux)

Contents

1.0. Software Installation

Follow the instructions under "Installing NightStar RT" on page 1 to install NightStar RT on your system. After the installation is complete, follow the instructions under "Obtaining License Keys" on page 3.

Then take a look at "Getting Started" on page 10 to find out how to access the *NightStar RT Tutorial* to learn about the features and functionality of the NightStar RT tools.

1.1. Prerequisites

Prerequisites for NightStar RT Version 3.2 for both the host system and target system are as follows:

1.1.1. Host System

Any of the following distributions:

- Red Hat Enterprise 3.0
- Red Hat Enterprise 4.0
- RedHawk 2.1 or higher

Any Intel Pentium Xeon or Intel EMT64 or AMD Opteron supported by the operating system distribution.

1.1.2. Target System

Any of the following distributions:

- RedHawk 2.1 or higher

Any Intel Pentium Xeon or Intel EMT64 or AMD Opteron system supported by RedHawk.

1.2. Installing NightStar RT

There are two methods of installing NightStar RT based on the installation media that you have.

- Network Installation
- CD Installation

1.2.1. Network Installation

Network installation is accomplished using NUU, Concurrent's Network Update and installation Utility.

If you do not have NUU installed on your system, visit the following web site and follow the instructions on installing and configuring NUU:

http://redhawk.ccur.com/updates

Once NUU is installed, you can install NightStar RT by invoking NUU as follows:

```
/usr/bin/nuu
```

1.2.1.1. Advanced Information

NUU is a graphical interface which uses the YUM and RPM subsystems to install and update software which is provided via a software repository.

Concurrent provides software repositories for many of its commercial products, including NightStar RT.

Any YUM-compatible client may be used to access the repositories.

If you installed and configured NUU via the instructions mentioned above, the NightStar RT repository is already configured and activated on your system.

However, if you prefer to use an alternative YUM client, you will need the following information:

The URL for the NightStar RT repository is:

```
https://redhawk.ccur.com/buffet?Login=login&Password=password&path=NightStar/RT/RedHawk/$basearch
```

where *login* and *password* should be replaced with your Login and Password assigned to you for the redhawk.ccur.com site.

1.2.2. CD Installation

To install NightStar RT using the *NightStar RT Installation CD*:

- Insert the *NightStar RT Installation CD* in the CD-ROM drive.

- If the CD does not auto-mount, mount the CD-ROM drive (assuming the standard mount entry for the CD-ROM device is **/media/cdrom** in **/etc/fstab**)

   ```
   mount /media/cdrom
   ```

- Double-click on the CD icon labeled NightStar RT on your desktop.

- Double-click on the icon labeled Launch Install Script.

NOTE

If you are not using a file browser to access the CD, change the current working directory to **/media/cdrom** and invoke the **./install-nstar** script.

- Double-click on the icon labeled Install Desktop Icons.

NOTE

Since you are running as root, these icons will only be installed on root's desktop. To install these on your normal user's desktop, run the following script when logged on as your normal user:

```
/usr/lib/NightStar-RT/bin/install-icons
```

1.3.   Obtaining License Keys

Permanent License Keys

- If you have purchased NightStar RT, you can obtain your permanent license keys
  at the following URL:

    http://www.ccur.com/NightStarRTKeys

  You will need your site ID, your email address, and your system identification number which was
  displayed during product installation. You can obtain that number again by running the following
  command on the system where the license keys will be installed:

    /usr/bin/nslm_admin --code

See "NightStar RT Licensing" on page 13 for more detailed information about the NightStar License
Manager (NSLM).

1.4.   Removing NightStar RT

To remove NightStar RT, mount the CD or ISO image as described in "CD Installation" on page 2 and
execute the following command as root:

    ./nstar-uninstall

2.0.   Documentation

Table 2-1 lists the NightStar RT 3.2 documentation available from Concurrent.

Table 2-1. NightStar RT Version 3.2 Documentation

Manual Name	Pub. Number
NightStar RT Installation Guide (Version 3.2)	0898008-3.2
NightStar RT Tutorial (Version 3.2)	0898009-060
NightProbe RT User's Guide (Version 3.1)	0898465-090
NightSim RT User's Guide (Version 3.5)	0898480-050
NightTrace RT User's Guide (Version 6.1)	0898398-150
NightTune RT User's Guide (Version 2.2)	0898515-020
NightView RT User's Guide (Version 6.3)	0898395-320
Data Monitoring Reference Manual	0890493-010
Quick Reference for shmdefine	0898010-030

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822.

Additionally, the manuals listed above are available:

- in PDF format in the **documentation** directory of the *NightStar RT Installation CD*

and, after installation:

- online using the **nhelp** utility

- in PDF format in the directory **/usr/lib/NightStar-RT/docs**

3.0. Overview of NightStar RT

The following sections describe the basic features of each of the NightStar RT tools.

3.1. NightProbe

The features of the NightProbe data monitor include:

- Non-intrusive sampling and recording of program data

- Synchronous and asynchronous data capture

- Flexible data display features

- Sampling, recording and replay APIs

- Time stamping of acquired data

NightProbe is a tool for independently monitoring, modifying and recording data values from multiple application resources, including programs, shared memory segments, and memory mapped files. NightProbe can be used in a development environment for debugging, analysis, proto-typing and fault injection, or in a production environment to create a GUI control panel for program input and output.

NightProbe utilizes a non-intrusive technique of mapping the target resource's address space into its own. Subsequent direct memory reads and writes by NightProbe allow it to sample and modify data without interrupting or otherwise affecting resources.

Synchronized and Asynchronous Logging

NightProbe can perform synchronous logging of data via a simple API. Asynchronous logging can be performed via on-demand sampling or a cyclic clock rate.

NightProbe provides for logging data items using tracepoints for simultaneous analysis by the NightTrace event analyzer. Sampled data can be combined with kernel trace and additional user trace data to obtain a synchronized picture of application and operating system behavior. NightProbe can record data to disk files or provide data directly to the NightTrace tool.

Interactive Sampling and Modification

NightProbe provides a flexible spreadsheet display for on-demand or cyclic sampling of data at user-specified refresh rates. Direct modification of user data is accomplished by typing in new values for data items into the spreadsheet. NightProbe provides colorized notification of violations of user-defined data thresholds for individual data items. NightProbe allows sampled data to be timestamped and passed off to user applications written with the NightProbe API for subsequent analysis, recording or customized display.

NightProbe supports scalar and structured data types in C/C++ and Fortran that have statically-determined addresses and shapes. NightProbe scans the symbol table and debug information of user programs allowing the user to browse for data items or specifically entering the names of data items to be monitored. Any application that contains symbol table and debug information may be used with NightProbe. No application source code changes are required.

3.2. NightSim

The features of the NightSim application scheduler include:

- Periodic execution of multiple processes

- Major and minor cycles with frame overrun notification and control

- Single point of scheduling control for distributed systems

- Ideal for simulation applications

NightSim is a tool for scheduling and monitoring time-critical applications that require predictable, cyclic process execution. Ideal for simulation applications, NightSim allows developers to dynamically adjust the execution of multiple, coordinated processes, their priorities, scheduling policies, and CPU assignments. With NightSim, users can monitor the performance of applications by displaying period execution times, minimums and maximums, and can take action when frame overruns occur.

NightSim provides a graphical interface to the operating system's Frequency-Based Scheduler (FBS), a high-resolution task scheduler that enables processes to run in cyclical patterns. NightSim allows users to easily configure groups of processes to run on local or distributed systems, and save the resulting configurations for reuse. A performance monitor gathers CPU utilization data for processes running under the FBS.

NightSim may be used during the development, debug and production phases of a simulation application. Simulation configurations can be saved as a script, which can then be executed to repeat a simulation. NightSim scripts are useful in target environments where GUI processing is prohibited or undesired. In addition, configuration files and scripts may be placed under any version control system.

Synchronized Distributed Scheduling

In addition to symmetric multiprocessors, NightSim supports multiple systems connected via Concurrent's Real-Time Clock and Interrupt Module. NightSim simplifies the creation of distributed scheduling and provides a single-point-of-control for managing the synchronized timing (start/stop/resume) of individual schedulers distributed across multiple target systems.

NightSim handles the interface to hardware such as real-time clocks and distributed interrupt sources. Users don't need to interface with the underlying operating system for scheduling operations.

Extensive Performance Statistics

NightSim monitors up to 11 different performance-related statistics as well as up to 15 additional parameters for each scheduled process. Using statistics such as minimum and maximum cycle times, users can optimize CPU utilization by balancing their load across multiple processors. NightSim displays are customizable, allowing users to select specific statistics and processes to monitor and the sorting criteria for weighted display.

3.3.   NightTrace

The features of the NightTrace event analyzer include:

- Synchronized graphical or text display of system application activity

- User-defined event logging in single or multi-threaded applications

- Kernel event logging including system calls, interrupts and exceptions

- Data analysis API

NightTrace is a tool for displaying and analyzing the dynamic behavior of applications, the Linux operating system and the interaction between them. NightTrace can log events from multiple processes executing simultaneously on multiple CPUs or systems. NightTrace can also combine user-defined application events with kernel events to present a synchronized view of the entire system. NightTrace then creates a graphical time-based view of all logged events. NightTrace allows users to zoom, search, filter, summarize and analyze events. Tracing analysis can be performed live or post execution.

NightTrace was specifically designed to meet the most stringent requirements of time-critical applications. Using synchronized, fast-access hardware clocks and kernel-free primitives, NightTrace tracepoints are logged with minimal overhead. Tracepoints can be inserted into device drivers, interrupt level code and any user application. Tracepoints can be left in production-quality applications even when not collecting trace data.

Graphical and Interactive

NightTrace graphically displays requested events and states along a timeline graph or event log to clearly show the relative timing of events and provide an overall picture of application and operating system activity. NightTrace can locate specific events and zoom in on them with a fine degree of granularity for precise timing observation. The NightTrace graphical display is completely user-configurable for customized viewing. Configurations can be saved and later recalled, and multiple configurations can be viewed simultaneously.

Kernel Trace Support

By combining system event information such as interrupts, exceptions, context switches, Linux system calls and device accesses together with event information from user applications, NightTrace provides a clear picture of the interaction between the kernel and user applications at any point during the application's run.

NightTrace provides statistical performance data about events and states, including frequency, time of occurrence, duration, gap and minimum and maximum times. Users can create state definitions and qualify events by specifying the applicable process, thread, CPU, system and event content. Conditional tracing can be expressed using C expression syntax. Displays can be customized to yield insight into operating system and application performance and behavior patterns.

NightTrace generates source code using an Analysis API that allows users to easily create custom applications that monitor or analyze application or system activity.

3.4.   NightTune

The features of the NightTune system and application tuner include:

- Dynamic display of system and application performance

- Monitoring of CPU use, memory paging and network operation

- Interactive control of processes, priorities, policies and interrupts

- Dynamic CPU affinity control for processes, threads and interrupts

NightTune provides a graphical interface to system facilities for monitoring and tuning application and system performance. Users can monitor the priority, scheduling policy, CPU assignment and CPU usage of user applications. NightTune also monitors system CPU usage, context switches, interrupts, memory paging and network activity.

NightTune can monitor processes individually or in groups determined by user or by CPU. NightTune also displays information about individual threads or tasks within a process. Multiple frames and windows are used to display information allowing users to customize their display.

Application Tuning

NightTune allows users to change the process attributes of an individual thread, task, process or group of processes as a whole using pop-up dialogs and drag-and-drop actions. For example, dragging a process icon to a CPU icon binds the process to that processor. The user then instantly sees the results of the tuning effort both graphically and as text.

System Tuning

NightTune allows users to change the CPU assignment of interrupts using pop-ups or drag-and-drop actions. NightTune optionally provides a textual log of all application and system tuning actions taking during a NightTune session.

3.5.   NightView

The features of the NightView source-level debugger include:

- Multi-system, multiprocessor, multi-process, multi-thread debugging via single interface

- Hot patches including breakpoints, monitorpoint and watchpoints

- Application speed conditions

- Dynamic memory "heap" debugging

- Modification and display of variables during execution

NightView allows users to simultaneously debug multiple, time-critical processes. With NightView, a programmer can change program execution and modify or display data without stopping or interrupting the program. Eventpoint conditions, such as hit and ignore counts, are patched directly into an application and can execute at full application speed. NightView provides fine-grained control without adversely affecting application timing.

NightView monitorpoints can display expressions at user-selected locations without stopping a process, thus providing data displays that are synchronized with the application's algorithms. Watchpoints utilize hardware address trap features that cause an application to stop when user-specified variables or memory locations are selectively read or modified.

Language-sensitive Debugging

NightView supports the debugging of multiple applications written in any combination of C/C++ and Fortran. All variables and expressions in each program are referenced in the appropriate language. NightView is also integrated with the NightTrace event analyzer. NightView can insert tracepoints at user-specified locations for concurrent or post execution analysis by NightTrace.

More Powerful Than The Gnu Debugger

NightView offers many features not available in the gnu debugger (`gdb`). Advantages of NightView include the ability for users to debug multiple processes from a single session and processes started from scripts. With NightView, patched-in code runs at full speed. While a process is executing, hot patching can modify variables or add eventpoints. Monitorpoints can display expressions and stack variables, and signals can be sent directly to the process, bypassing the debugger.

Dynamic Memory Debugging

NightView includes an interactive memory debugger that helps find and eliminate memory problems during the debug process without code recompilation. NightView watches for heap memory leaks, monitors the amount of memory an application uses, and tracks how it allocates and frees memory. With its memory debugger enabled, NightView lets users track heap allocations and deallocations in real-time, thus allowing for more efficient debugging than post-run analysis. Programmers can stop execution, check for problems, test patches and then continue debugging. NightView can detect double-frees, dangling pointers, heap area overruns, and other common user application bugs.

3.6.   Datamon

Datamon is a user application interface that allows user programs to monitor, record, and modify variables in independently executing processes in real-time. It includes the ability to scan a program file for eligible variables and obtain detailed information about their attributes, including type name, atomic type, bit size, bit offset, shape, component members, and address. Datamon utilizes a non-intrusive technique for accessing and modifying variables.

3.7.   Shmdefine

Shmdefine aids in the sharing of data between independent programs. While most useful for sharing common blocks between Fortran programs, it helps Fortran, C, and Ada programs to effectively utilize the IPC shared memory services.

4.0. Getting Started

The *NightStar RT Tutorial* is **highly recommended** as an introduction to the NightStar RT product. This tutorial integrates all of the NightStar RT tools into one cohesive example incorporating various scenarios which demonstrate their extensive functionality.

The tutorial is available in PDF format in the **documentation** directory of the *NightStar RT Installation CD* as well as in **/usr/lib/NightStar-RT/docs** after installation.

The HyperHelp™ online version of the tutorial can be accessed by double-clicking on the NightStar RT Documentation icon installed on the desktop and selecting the NightStar RT Tutorial from the Bookshelf.

In addition, the tutorial can be launched from the Help menu of any NightStar RT tool or can be started by issuing the following command:

```
nhelp nstar-rt-tutorial
```

from the command line.

4.1. Capabilities

Most operations with NightStar RT do not require any special privileges. However, if you wish to take full advantage of NightStar RT capabilities without running as the root user, additional configuration steps are required.

Linux provides a means to grant otherwise unprivileged users the authority to perform certain privileged operations. The Pluggable Authentication Module (see **pam_capability(8)**) is used to manage sets of capabilities, called *roles*, required for various activities.

Table 4-2 lists the advantages granted to non-root users with the capabilities suggested for use with NightStar RT:

Table 4-2. Capabilities and their Effects

Capability	Advantage
CAP_IPC_LOCK	Allows NightTrace to lock critical pages into memory related to User Trace event buffers.
CAP_SYS_RAWIO	Allows NightProbe to gain access to PCI devices and memory mapped system files, such as **/dev/mem**.
CAP_SYS_NICE	Allows the NightStar RT tools to set the scheduling policy, scheduling priority, and CPU affinity of processes. Allows NightTune to set the CPU affinity of interrupt and to shield CPUs from process, interrupts. and hyper-threading interference.

Linux systems should be configured with an *nstaruser* role which provides the CAP_SYS_NICE, CAP_SYS_RAW_IO and CAP_IPC_LOCK capabilities.

Edit **/etc/security/capability.conf** and define the nstaruser role (if it is not already defined) in the "ROLES" section:

```
role nstaruser CAP_SYS_NICE CAP_IPC_LOCK CAP_SYS_RAWIO
```

Additionally, for each NightStar RT user on the target system, add the following line at the end of the file:

```
user  username    nstaruser
```

where *username* is the login name of the user.

If the user requires capabilities not defined in the nstaruser role, add a new role which contains nstaruser and the additional capabilities needed, and substitute the new role name for nstaruser in the text above.

In addition to registering your login name in **/etc/security/capability.conf**, certain files under the **/etc/pam.d** directory must also be configured to allow capabilities to be activated.

To activate capabilities, add the following line to the end of selected files in **/etc/pam.d** if it is not already present:

```
session required pam_capability.so
```

The list of files to modify is dependent on the list of methods that will be used to access the system. The following table presents a recommended configuration that will grant capabilities to users of the services most commonly employed in accessing a system.

Table 4-3. Recommended /etc/pam.d Configuration

/etc/pam.d File	Affected Services	Comment
remote	telnet rlogin rsh (when used w/o a command)	Depending on your system, the **remote** file may not exist. Do not create the **remote** file, but edit it only if it is present.
login	local login (e.g. console) telnet* rlogin* rsh* (when used w/o a command)	*On some versions of Linux, the presence of the **remote** file limits the scope of the **login** file to local logins. In such cases, the other services listed here with **login** are then affected solely by the **remote** configuration file.
rsh	rsh (when used with a command)	e.g. **rsh system_name a.out**
sshd	ssh	You must also edit **/etc/ssh/sshd_config** and ensure that the following line is present: **UsePrivilegeSeparation no**
gdm	gnome sessions	
kde	kde sessions	

If you modify **/etc/pam.d/sshd** or **/etc/ssh/sshd_config**, you must restart the **sshd** service for the changes to take effect:

```
/sbin/service sshd restart
```

In order for the above changes to take effect, the user must log off and log back onto the target system.

NOTE

To verify that you have been granted capabilities, issue the following command:

```
/usr/sbin/getpcaps $$
```

The output from that command will list the roles currently assigned to you.

5.0.   NightStar RT Licensing

NightStar RT uses the NightStar License Manager (NSLM) to control access to the NightStar RT tools.

License installation requires a licence key provided by Concurrent. The NightStar RT tools request a licence (see "License Requests" on page 14) from a license server (see "License Server" on page 14).

Two license modes are available, fixed and floating, depending on which product option you purchased. Fixed licenses can only be served to NightStar RT users from the local system. Floating licenses may be served to any NightStar RT user on any system on a network.

Tools are licensed per system, per concurrent user. Concurrent usage of any or all NightStar RT tools by the same user from the same system automatically share a single license. The intent is to allow *n* developers to fully utilize all the tools at the same time while only requiring *n* licenses. When operating the tools in remote mode, where a tool is launched on a local system but is interacting with a remote system, licenses are required only from the host system.

You can obtain a license report which lists all licenses installed on the local system, current usage, and expiration date for demo licenses (see "License Reports" on page 14).

The default operating system configuration may include a strict firewall which may interfere with floating licenses. See "Firewall Configuration for Floating Licenses" on page 15 for information on handling such configurations.

5.1.   License Keys

Licenses are granted to specific systems to be served to either local or remote clients, depending on the license model, fixed or floating.

License installation requires a license key provided by Concurrent. To obtain a license key, you must provide your system identification code. The system identification code is generated by the **nslm_admin** utility:

```
nslm_admin --code
```

System identification codes are dependent on system configurations. Reinstalling Linux or NightStar RT on a system or replacing network devices may require you to obtain new license keys.

To obtain a license key, use the following URL and click on the *Licenses* link:

http://www.ccur.com/NightStarRTKeys

Provide the requested information, including the system identification code. Your license key will be immediately emailed to you.

Install the license key using the following command:

```
nslm_admin --install=xxxx-xxxx-xxxx-xxxx-xxxx
```

where *xxxx-xxxx-xxxx-xxxx-xxxx* is the key included in the license acknowledgment email.

If the required information is not readily available, or you have special circumstances, contact Concurrent support (see "Direct Software Support" on page 20 for more information).

5.2.   License Requests

By default, the NightStar RT tools request a license from the local system. If no licenses are available, they broadcast a license request on the local sub-net associated with the system's hostname.

You can control the license requests for an entire system using the `/etc/nslm.config` configuration file.

By default, the `/etc/nslm.config` file contains a line similar to the following:

```
:server @default
```

The argument `@default` may be changed to a colon-separated list of system names, system IP addresses, or broadcast IP addresses. Licenses will be requested from each of the entities found in the list until a license is granted or all entries in the list are exhausted.

For example, the following setting prevents broadcast requests for licenses by only specifying the local system:

```
:server localhost
```

The following setting requests a license from `server1`, then `server2`, and then a broadcast request if those fail to serve a license:

```
:server server1:server2:129.134.30.0
```

Similarly, you can control the license requests for individual invocations of the tools using the `NSLM_SERVER` environment variable. If set, it must contain a colon-separated list of system names, system IP addresses, or broadcast IP addresses as described above. Use of the `NSLM_SERVER` environment variable takes precedence over settings defined in `/etc/nslm.config`.

5.3.   License Server

The NSLM license server is automatically installed and configured to run when you install NightStar RT.

The `nslm` service is automatically activated for run levels 2, 3, 4, and 5. You can check on these settings by issuing the following command:

```
/sbin/chkconfig --list nslm
```

In rare instances, you may need to restart the license server via the following command:

```
/sbin/service nslm restart
```

See `nslm(1)` for more information.

5.4.   License Reports

A license report can be obtained using the `nslm_admin` utility.

```
nslm_admin --list
```

lists all licenses installed on the local system, current usage, and expiration date (for demo licenses). Use of the `--verbose` option also lists individual clients to which licenses are currently granted.

Adding the **--broadcast** option will list this information for all servers that respond to a broadcast request on the local sub-net associated with the system's hostname.

See **nslm_admin(1)** for more options and information.

5.5. Firewall Configuration for Floating Licenses

The default Red Hat configuration includes a strict firewall which interferes with floating licenses.

If such a system is used to serve licenses, then at least one port must be opened in its firewall to allow server requests to pass. See "Serving Licenses with a Firewall" on page 15 for more information.

Similarly, if such a system is host to the NightStar RT tools, then at least one port must be opened in its firewall so that it can receive licenses from the license server. If this is not done, a tool requesting a floating license will not receive it and will not function properly. See "Running NightStar RT Tools with a Firewall" on page 16 for more information.

5.5.1. Serving Licenses with a Firewall

Following are a few approaches for allowing the NSLM license server to serve floating licences when the system on which it is running is configured with a firewall:

- disable the firewall on the system entirely
- allow NSLM license requests from a specific system (or one of several)
- allow NSLM license requests from any system on a particular subnet (or one of several)
- allow NSLM license requests from any system

NOTE

You must be root in order to modify the firewall configuration.

To disable the firewall entirely, execute:

```
service iptables stop
```

and then remove the **/etc/sysconfig/iptables** file:

```
rm -f /etc/sysconfig/iptables
```

This option may not be as dangerous as it seems. Often, whole networks are protected with a firewall so it is not necessary for individual systems on the network to be protected further. If unsure, check with your network administrator.

For the remaining cases, a simple modification should be made to the **/etc/sysconfig/iptables** file. By default, that file should contain a line like the following:

```
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

To allow NSLM license requests from a specific system, insert the following lines before the REJECT line:

```
-A RH-Firewall-1-INPUT -p udp -m udp -s system --dport 25517 -j ACCEPT
```

```
                    -A RH-Firewall-1-INPUT -p tcp -m tcp -s system --dport 25517 -j ACCEPT
```

Those lines can be repeated for multiple systems.

To allow NSLM license requests from any system on a particular subnet, insert the following lines before the REJECT line:

```
        -A RH-Firewall-1-INPUT -p udp -m udp -s subnet/mask --dport 25517 -j ACCEPT
        -A RH-Firewall-1-INPUT -p tcp -m tcp -s subnet/mask --dport 25517 -j ACCEPT
```

The subnet might be of a form like 192.168.1.0 and the mask could be a traditional network mask like 255.255.255.0 or a single number like 24, which indicates the number of bits from the left that are part of the mask. For example, 192.168.1.0/255.255.255.0 and 192.168.1.0/24 are equivalent.

Those lines can be repeated for multiple subnets.

To allow NSLM license requests from any system, insert the following lines before the REJECT line:

```
        -A RH-Firewall-1-INPUT -p udp -m udp --dport 25517 -j ACCEPT
        -A RH-Firewall-1-INPUT -p tcp -m tcp --dport 25517 -j ACCEPT
```

After modifying **/etc/sysconfig/iptables**, execute:

service iptables restart

5.5.2.   Running NightStar RT Tools with a Firewall

Following are a few approaches for allowing a NightStar RT tool to receive floating licenses from a license server, when the system running the NightStar RT tool is configured with a firewall:

- disable the firewall on the requesting system entirely
- allow NSLM licenses from a specific license server (or one of several)
- allow NSLM licenses from any system on a particular subnet (or one of several)
- allow NSLM licenses from any system

NOTE

You must be root in order to modify the firewall configuration.

To disable the firewall entirely, execute:

service iptables stop

and then remove the **/etc/sysconfig/iptables** file:

rm -f /etc/sysconfig/iptables

This option may not be as dangerous as it seems. Often, whole networks are protected with a firewall so it is not necessary for individual systems on the network to be protected further. If unsure, check with your network administrator.

For the remaining cases, a simple modification should be made to the **/etc/sysconfig/iptables** file. By default, that file should contain a line like the following:

NightStar RT Installation Guide Version 3.2

```
          -A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

To allow NSLM licenses from a specific system running a license server, insert the following line before the REJECT line:

```
          -A RH-Firewall-1-INPUT -p udp -m udp -s server --sport 25517 -j ACCEPT
```

That line can be repeated for multiple servers.

To allow NSLM licenses from any system running a license server on a particular subnet, insert the following before the REJECT line:

```
          -A RH-Firewall-1-INPUT -p udp -m udp -s subnet/mask --sport 25517 -j ACCEPT
```

The subnet might be of a form like 192.168.1.0 and the mask could be a traditional network mask like 255.255.255.0 or a single number like 24, which indicates the number of bits from the left that are part of the mask. For example, 192.168.1.0/255.255.255.0 and 192.168.1.0/24 are equivalent.

That line can be repeated for multiple subnets.

To allow NSLM licenses from any system running a license server, insert the following line before the REJECT line:

```
          -A RH-Firewall-1-INPUT -p udp -m udp --sport 25517 -j ACCEPT
```

After modifying **/etc/sysconfig/iptables**, execute:

service iptables restart

Following are a few approaches for allowing the NSLM license server to serve floating licences when the system on which it is running is configured with a firewall:

- disable the firewall on the system entirely

- allow NSLM license requests from a specific system (or one of several)

- allow NSLM license requests from any system on a particular subnet (or one of several)

- allow NSLM license requests from any system

NOTE

You must be root in order to modify the firewall configuration.

To disable the firewall entirely, execute:

service iptables stop

5.6. License Support

For additional aid with licensing issues, contact the Concurrent Software Support Center. See "Direct Software Support" on page 20 for details.

6.0. Architecture Interoperability

The NightStar RT tools were designed to be used in a self-hosted environment as well as remotely, separating the host processing from the time-critical target system.

The following table describes the interoperability of each tool between systems of different architectures.

NightStar Tool	Host	Target	Limitation
NightProbe	Intel	AMD64	No limitations
	AMD64	Intel	No limitations
NightSim	Intel	AMD64	No limitations
	AMD64	Intel	No limitations
NightTrace	Intel	AMD64	Not supported
	AMD64	Intel	Not supported
NightTune	Intel	AMD64	No limitations
	AMD64	Intel	No limitations
NightView	Intel	AMD64	Cannot debug 32-bit applications on AMD64 systems
	AMD64	Intel	Cannot execute AMD64 user applications on Intel systems

7.0. Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time.

You may also submit a request for assistance at any time by using the Concurrent Computer Corporation web site at **http://www.ccur.com/isd_support_contact.asp** or by sending an email to **support@ccur.com**.