



Trusted RUBIX™

Version 6

Trusted RUBIX

Installation and Quick Start Guide

Red Hat Enterprise Linux 6 SELinux i386/i686 Platform

Revision 1

RELATIONAL DATABASE MANAGEMENT SYSTEM

Infosystems Technology, Inc.

4 Professional Dr - Suite 118

Gaithersburg, MD 20879

TEL +1-202-412-0152

© 1981 - 2011 Infosystems Technology, Inc. ("ITI"). All rights reserved. Unpublished work. Commercial computer software and software documentation: Government users are subject to ITI's standard license agreement per DFARS 227.7203-3 or, in non-DoD agencies where such protection is unavailable, to "restricted rights" under applicable FAR System clauses.

Infosystems Technology, Inc.
4 Professional Dr - Suite 118
Gaithersburg, MD 20879

THIS DOCUMENTATION CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF INFOSYSTEMS TECHNOLOGY, INC. USE, DISCLOSURE, OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF INFOSYSTEMS TECHNOLOGY, INC. FOR FULL DETAILS OF THE TERMS AND CONDITIONS FOR USING THE SOFTWARE, PLEASE REFER TO THE ITI-TRUSTED RUBIX USER LICENSE AGREEMENT.

The information in this document is subject to change without notice and should not be construed as a commitment by ITI.

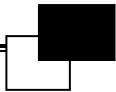
Infosystems Technology, Inc. assumes no responsibility for any errors that may appear in this document.

RUBIX® is a trademark of Infosystems Technology, Inc.

UNIX® is a trademark of The Open Group.

Microsoft® is a trademark of the Microsoft Corporation.

Printed in U.S.A.



OVERVIEW 1

INSTALLATION 1

 OPERATING SYSTEM INSTALLATION.....2

 TRUSTED RUBIX 6.0 INSTALLATION3

CREATING A DEVELOPMENT USER 4

CONFIGURING LOCAL COMMUNICATIONS..... 7

CONFIGURING REMOTE COMMUNICATIONS 7

UNINSTALLING TRUSTED RUBIX..... 10

EXECUTION AND USE 11

 ASSUMING A TRUSTED RUBIX ROLE.....11

 STARTING THE TRUSTED RUBIX DISPATCHER11

 CREATING A DATABASE12

 PERFORMING CLIENT SQL OPERATIONS12

 CONFIGURING AND CREATING ODBC APPLICATIONS12

OVERVIEW OF USER DOCUMENTATION..... 13

SUPPORT AND FURTHER INFORMATION..... 13

Overview

This document describes the installation procedures and basic operation of the Trusted RUBIX DBMS software. It applies to installing Trusted RUBIX 6.0 on the Red Hat Enterprise Linux 6 (RHEL6) operating system on i386 or i686 hardware. The procedures and processes described here apply only for version 6.0 of the Trusted RUBIX software.

Installation

The installation must be performed on the RHEL6 SELinux operating system running on i386 or i686 hardware. Configuration of the operating system and installation of Trusted RUBIX must be performed as the Linux *root* user. Furthermore, the SELinux security policy should be set to permissive mode or the user must assume the *sysadm_r* role. The easiest and often most failsafe method is simply to place the SELinux policy into permissive mode and become the *root* user. In permissive mode the SELinux policy enforcement is disabled and any operation that would have been denied due to the SELinux policy is audited. The SELinux policy may be placed into permissive mode using the GUI SELinux Management tool. It may be started from the *System -> Administration -> SELinux Management* menu. Permissive mode should only be used for development and configuration and not for deployment.

The SELinux policy on RHEL6 must either be operating with the Targeted policy or the MLS policy. While Trusted RUBIX enforces its SELinux behavior in the same way with either policy, the operating system will be easier to use operating with the Targeted policy.

The Targeted policy applies SELinux Type Enforcement (TE) to only a subset of system services. Typical user subjects run in unconfined modes. It also applies Multi-Category Security (MCS). Multi-Category Security is similar to Multi-Level Security except it has only a single sensitivity level and multiple categories. The Targeted policy is the default policy configuration for RHEL6 and tends to be less problematic. It is strongly recommended to use RHEL6 / Trusted RUBIX with the SELinux Targeted policy until becoming very familiar with SELinux policy behavior. It is possible to switch between the Targeted and MLS policies using the SELinux Management tool; however, databases created under one policy will not function under the other policy.

The MLS policy applies SELinux Type Enforcement to every subject and object of the operating system. SELinux TE behavior denies any operation unless an explicit TE rules allows it. Therefore, with the MLS policy all desired operations must have corresponding SELinux policy rules. For this reason, the MLS policy and its behavior can be complex.

The Trusted RUBIX SELinux Guide has more information about SELinux and its relationship to Trusted RUBIX.

The book [*SELinux by Example*](#) (ISBN: 0-13-196369-4) provides a good, but somewhat outdated, introduction to using and configuring SELinux.

For more information about the SELinux policy on RHEL6 see the following URL:

https://access.redhat.com/knowledge/docs/Red_Hat_Enterprise_Linux/

General information about SELinux, including an active mailing list, may be found at:

<http://www.nsa.gov/research/selinux/>

Operating System Installation

Install the RHEL6 operating system on your hardware and apply all available updates. Ensure that you install the version of RHEL6 that corresponds to your version Trusted RUBIX.

It is highly recommended that you apply all updates to the operating system at this time. You may update your operating system's software by using the tool found under the *System->Administration->Update System* menu. After the installation has completed, additional packages must be installed. First, become the *root* user and either assume the *sysadm_r* role or place the SELinux policy into permissive mode.

The *yum install* operation may be used to download and install a package, as shown below. Note that using the *yum install* operation will ensure that all dependent packages are downloaded and installed as well as the target package.

```
yum install PACKAGE_NAME
```

Information about a package, including its installation status, may be found using the following command:

```
yum info PACKAGE_NAME
```

An Internet connection is required to perform the *yum* operations and the command will automatically locate the proper package repository on the Internet. If you do not have a network connection on the host machine you will need to manually download the package files and all dependent packages on another machine and then transfer them to the host machine.

Installing an individual package file that resides on the host machine may be accomplished with the following command (from the directory containing the package file):

```
rpm -ivh ./RPM_PACKAGE_FILE
```

Install all of the following packages. Depending on the configuration of your operating system installation, some of these packages may already be installed; if so, you will receive an appropriate message during the package installation operation.

- selinux-policy
- selinux-policy-targeted
- selinux-policy-mls
- selinux-policy-devel (needed for policy development only, may not be available on all platforms)
- netlabel_tools
- policycoreutils
- policycoreutils-newrole (not available on all OS versions)
- libselinux
- libcap
- zlib
- openssl
- unixODBC (only if platform will run ODBC clients)

→ unixODBC-devel (only if platform will be used for ODBC client development)

Trusted RUBIX 6.0 Installation

Place the Trusted RUBIX 6.0 installation disk into your RHEL6 computer. Note that a copy of this installation guide is included on the disk. Copy the following files to your computer (the *x*'s will be replaced by your actual version numbers):

- `rubixdbms-6.0.x-x.el6.ixxx.rpm` (server machine only)
- `rubixdbms-devel-6.0.x-x.el6.ixxx.rpm` (server/development machine only)
- `rubixdbms-doc-6.0.x-x.el6.ixxx.rpm` (any machine for which you want the documentation installed)
- `rubixdbms-odbc-6.0.x-x.el6.ixxx.rpm` (client machines without the `rubixdbms-devel` package installed)

Become the `root` user and either assume the `sysadm_r` role or place the SELinux policy into permissive mode. If you are upgrading from a previous version of Trusted RUBIX, it is recommended that you first uninstall the old version. Your data files will not be removed and a backup of the `rxconfig` file will be automatically saved by the `rpm` tool.

This version of Trusted RUBIX is **not backwards compatible** with databases created with previous versions. Therefore, only databases created with the new version will function with this version of Trusted RUBIX. You may request a database conversion tool that will convert between different version of Trusted RUBIX. To receive the conversion tool, please email ITI at support@rubix.com with the version numbers of both the new version of Trusted RUBIX and the version of Trusted RUBIX used to create your existing database. You may retrieve the version number of an installed version of Trusted RUBIX by issuing one of the following commands:

```
rpm -qi rubixdbms
```

```
yum info rubixdbms
```

To uninstall a package issue the following command:

```
rpm -e PACKAGE_NAME
```

The Trusted RUBIX package names for server platforms are `rubixdbms`, `rubixdbms-devel`, and `rubixdbms-doc`. All client-only platforms and any server platform which does not have the `rubixdbms-devel` package installed may obtain the ODBC client libraries from the `rubixdbms-odbc` package.

For each platform that will be a Trusted RUBIX **server**, install each package file in the order given above using the following commands:

```
cd RPM_PACKAGE_FILE_DIR
rpm -ivh ./RPM_PACKAGE_FILE
```

These steps will install packages named `rubixdbms`, `rubixdbms-devel`, `rubixdbms-doc`, and `rubixdbms-odbc`. The install places Trusted RUBIX into the `/var/lib/RUBIXdbms` directory. All user executable programs are accessible from `/usr/bin` as soft links. The ODBC library is accessible from `/usr/lib`. The install creates the `rubix` user and the `rubixtp` group, which are used to isolate Trusted RUBIX files and processes. All user documentation is accessible in the `/var/lib/RUBIXdbms/doc` directory.

The install creates two SELinux policy modules:

- *rubix-base* and
- *rubix-dev*

The *rubix-base* module contains all of the policy rules needed for Trusted RUBIX to operate within the RHEL6 operating system and base rules used by the *rubix-dev* policy module. Source code is not included for the *rubix-base* policy module and may not be altered by the end user. The *rubix-dev* policy module contains site-specific SELinux policy for DBMS objects. The install process places the source code for the *rubix-dev* policy into the `/var/lib/RUBIXdbms/etc/selinux` directory. It is intended to be used for on-site custom policy development. The default *rubix-dev* policy module contains policy rules that represent the possible uses of SELinux with Trusted RUBIX and serves to demonstrate its capabilities. For more information on developing custom security policies for Trusted RUBIX please see the *Trusted RUBIX SELinux Guide*.

For each platform that will be used as a Trusted RUBIX ODBC client and **does not** have the *rubixdbms-devel* package installed, install the ODBC client package as follows:

```
cd RPM_PACKAGE_FILE_DIR
rpm -ivh ./rubixdbms-odbc-6.0.x-x.xxx.ixxx.rpm
```

Take note to install the ODBC client package for your particular client machine. Client packages are located in the *Clients* directory of your installation disk. Packages with the *fcxx* designation are for Fedora, packages with the *el5* designation are for RHEL5, and packages with the *el6* designation are for RHEL6. Note that the *rubixdbms-devel* package includes the ODBC libraries for a server machine, so only install the *rubixdbms-odbc* package on a server machine if the *rubixdbms-devel* package is not installed.

For information on installing the ODBC driver on Microsoft Windows please see the Trusted RUBIX ODBC Guide. Note that the Microsoft Visual C++ 2008 Redistributable Package must be installed on any Windows platform that will host the Trusted RUBIX ODBC Driver. Installation instructions and the package may be found at:

<http://www.microsoft.com/Downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en>

Creating a Development User

When performing development of Trusted RUBIX client applications it may be beneficial to have a logon user that may access all of the Trusted RUBIX roles. This section describes how to create such a development user. The development user will be able to reach all roles created by the default *rubix-dev* policy. Note that the *rubix-dev* policy has role transition rules that allow all roles (even client roles) to be reached from *staff_r*. For a production environment, it is recommended to partition the roles between users as the security requirements dictate. For information on how to create users for a production environment please see the Trusted RUBIX SELinux Guide. **For the following steps the *MLS* policy is assumed. If you are using the *Targeted* policy please substitute the *s0-s0:c0.c1023* level range for the given *s0-s15:c0.c1023* level ranges.** If you cut-and-paste any of these commands, note that the double quote character often does not translate properly and must be re-typed. The steps that follow may be used to create a Linux user that may access all of the Trusted RUBIX roles:

1. Create the Linux user *rxdev* (*useradd* command)

2. Create SELinux user *rxdev_u* with a level range of *s0-s15:c0.c1023* (*s0-s0:c0.c1023* for *Targeted* policy) and all of the following roles: *staff_r*, *rubix_dbadm_r*, *rubix_op_r*, *rubix_auditadm_r*, *rubix_secadm_r*, *objset1_rubix_adm_r*, *rubix_client_r*, *rubix_remote_client_r*. The ‘\’ character indicates a line continuation and should not be typed as part of the command.

```
semanage user -a -r s0-s15:c0.c1023 -R "staff_r rubix_dbadm_r rubix_op_r \
rubix_auditadm_r rubix_secadm_r objset1_rubix_adm_r rubix_client_r rubix_remote_client_r" \
rxdev_u
```

3. Map the *rxdev* Linux user to the *rxdev_u* SELinux user.

```
semanage login -a -s rxdev_u -r s0-s15:c0.c1023 rxdev
```

4. Make the *staff_r* role the default role for the *rxdev_u* user by adding the following lines to the */etc/selinux/targeted/contexts/default_contexts* and */etc/selinux/mls/contexts/default_contexts* files. Note that part of these lines may already exist in which case you will simply have to append *staff_r:staff_t:s0* to the end of each designated line.

```
system_r:local_login_t:s0    user_r:user_t:s0 staff_r:staff_t:s0
system_r:remote_login_t:s0  user_r:user_t:s0 staff_r:staff_t:s0
system_r:sshd_t:s0         user_r:user_t:s0 staff_r:staff_t:s0
system_r:xdm_t:s0          user_r:user_t:s0 staff_r:staff_t:s0
```

5. Adds default types for each Trusted RUBIX role by adding the following lines to the */etc/selinux/targeted/contexts/default_type* and */etc/selinux/mls/contexts/default_type* files. The *staff_r:staff_t* entry may already exist.

```
staff_r:staff_t
rubix_dbadm_r:rubix_dbadm_t
rubix_op_r:rubix_op_t
rubix_auditadm_r:rubix_auditadm_t
rubix_secadm_r:rubix_secadm_t
objset1_rubix_adm_r:objset1_rubix_adm_t
rubix_client_r:rubix_client_t
rubix_remote_client_r:rubix_remote_client_t
```

6. Login as the *rxdev* user

7. Add the terminal type as a trusted type so it can use *newrole*

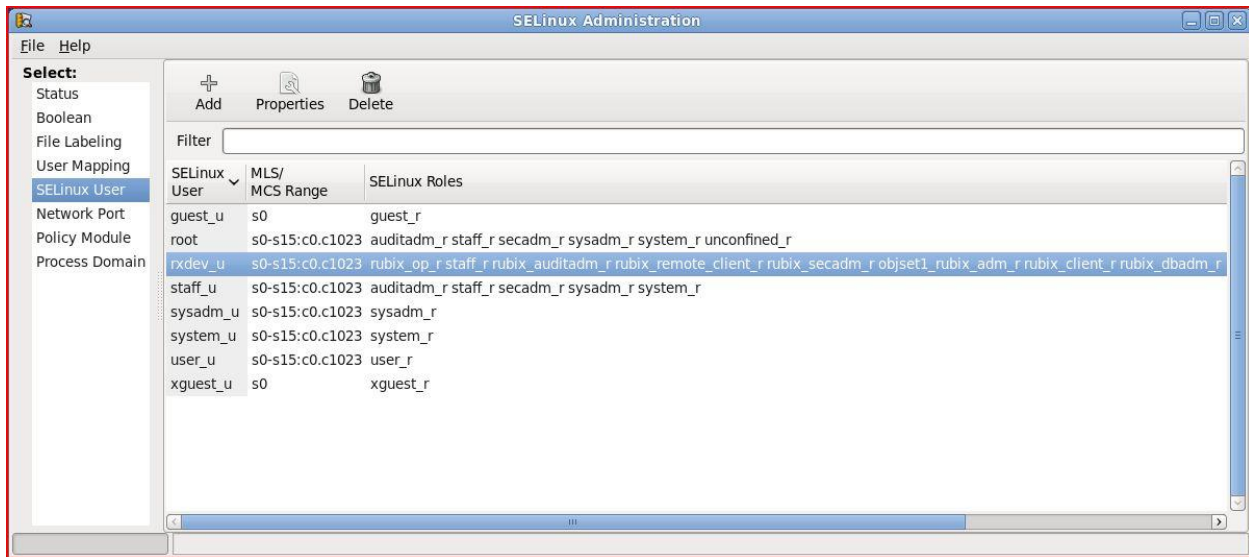
```
ls -Z `tty`
```

add the corresponding type (e.g., *user_devpts_t*) to both the */etc/selinux/targeted/contexts/securetty_types* and */etc/selinux/mls/contexts/securetty_types* files.

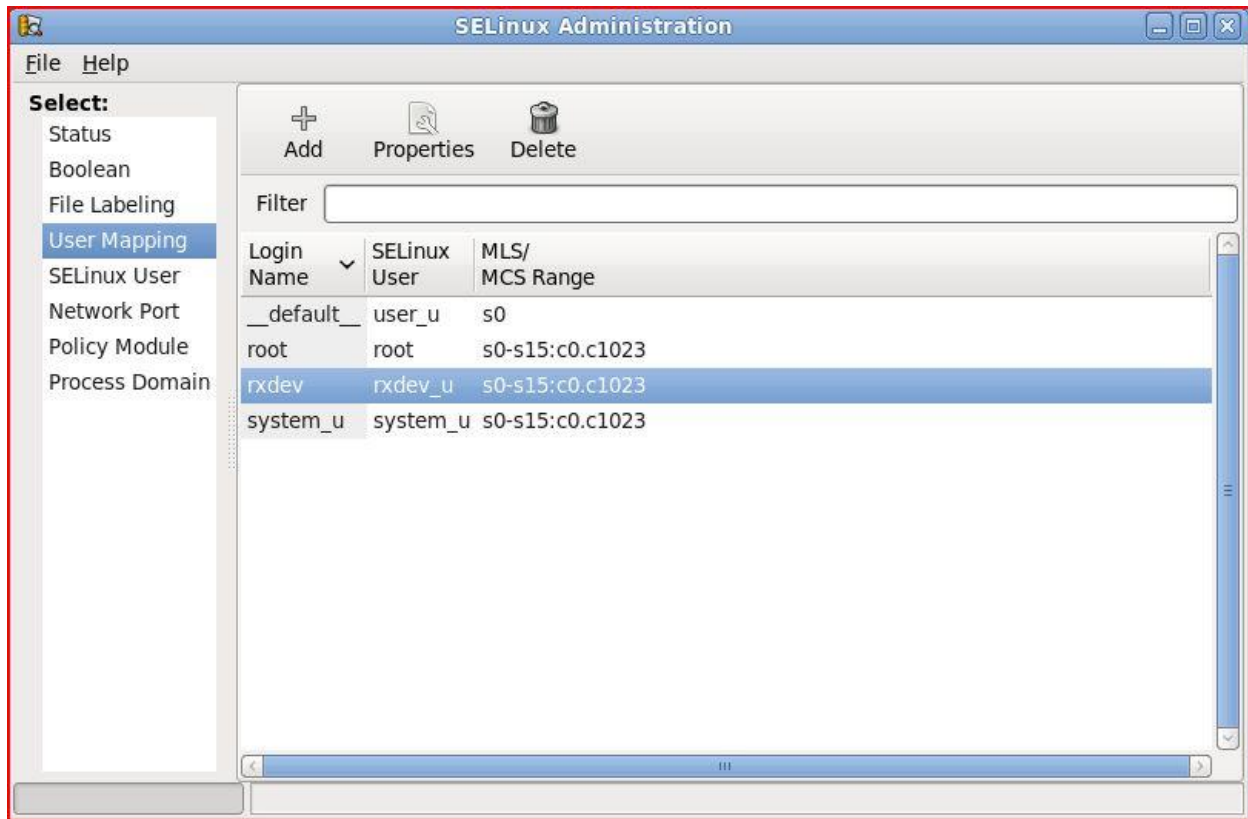
8. Change to the desired role (e.g., *newrole -r rubix_dbadm_r*) and use Trusted RUBIX. You may verify your current security context using the *id -Z* command.

Many of the preceding steps may also be accomplished using the SELinux Administration GUI tool. It may be found under the “System->Administration->SELinux Management” menu.

The `rxdev_u` SELinux user configuration will look like the following:



The *rxdev* Linux user to *rxdev_u* SELinux user mapping will look like the following:



Configuring Local Communications

Local connections use UNIX sockets and are labeled automatically by RHEL6 with the label of the client's process. Therefore, no further configuration is needed. UNIX socket files are created in the */var/lib/RUBIXdbms/sockets* directory.

To verify a correct installation, it is recommended that Trusted RUBIX first be used in local communications mode before attempting to use Trusted RUBIX in remote communications mode. To use Trusted RUBIX in local communications mode, simply execute your client program (e.g., *rxisql*) on the same machine that contains the Trusted RUBIX server and do not specify a remote host name in the database connection string (e.g., use *MyDB* as opposed to *MyDB@remote.host.com*).

Configuring Remote Communications

The Trusted RUBIX software uses a single port to communicate with remote clients via INET sockets. The default port number is 4156. If a different port number is desired it may be configured in the */var/lib/RUBIXdbms/etc/rxconfig* file. If a port other than 4156 is desired, the *dispatcher.listenport* entry in the *rxconfig* file should be added (if it does not exist) or modified as follows:

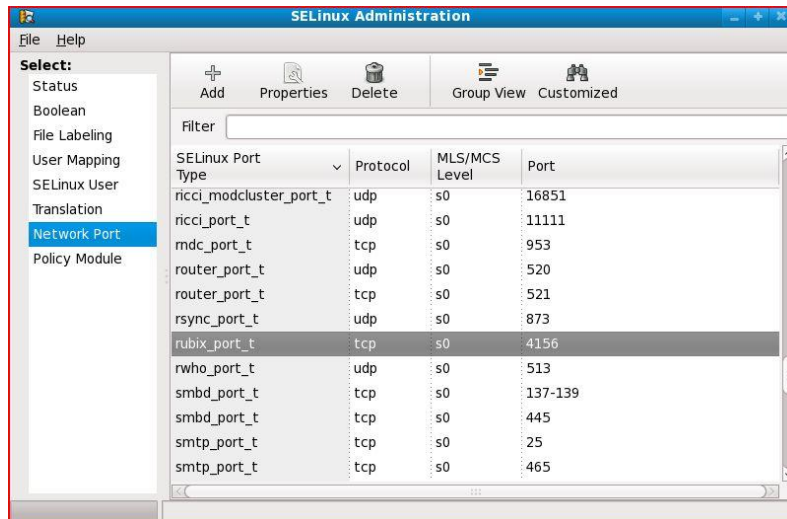
```
dispatcher.listenport      = DSPR_PORT_NUMBER
```

If the port number is set to 0 (the number zero) then remote connections to Trusted RUBIX will be disabled. In this configuration only clients on the local host machine will be able to connect to Trusted RUBIX servers. Note that if a port other than the default is used, the port number must be explicitly passed to remote Trusted RUBIX client programs during connection. Local Trusted RUBIX client programs use UNIX sockets and do not require a port number.

For remote connections the port number must be configured for SELinux using the SELinux Management GUI tool or the *semanage* command. The GUI tool may be started from the *System -> Administration -> SELinux Management* menu. Once started choose “Network Port” and add a new port with the following values:

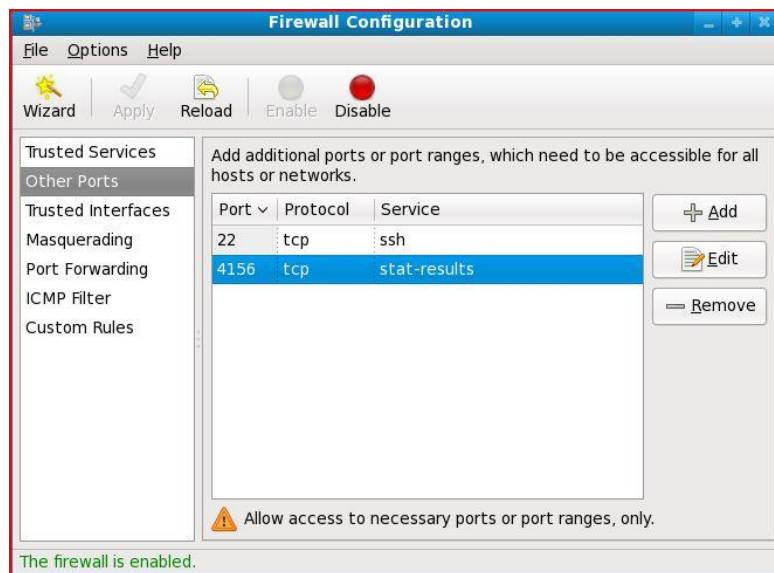
Port Number: Your port number (e.g., 4156)
Protocol: tcp
SELinux Type: rubix_port_t
MLS/MCS Level: s0

The Network Port configuration will look as follows:



The firewall configuration must also allow communication access to the port. This may be accomplished through the GUI Firewall tool from the *System -> Administration -> Firewall* menu. Click on “Other Ports” and then “Add.” Check “User Defined” and add the desired port number using the *tcp* protocol. Click “OK” and then *File -> Apply*.

The Firewall Configuration will look as follows:



Trusted RUBIX extracts the remote user’s SELinux context directly from the context of the remote socket connection. Therefore, socket connections must be configured with an appropriate SELinux label. This may be accomplished by using *netlabel* or *IPSec*. For more information please see the Trusted RUBIX SELinux Guide and the corresponding RHEL6 documentation.

By default, RHEL6 leaves socket connections from remote hosts unlabeled. Trusted RUBIX assigns the

system_u:object_r:unlabeled_t:s0 context to sessions using unlabeled socket connections. The default *rubix-dev* policy does not permit sessions with the *unlabeled_t* type to connect to a Trusted RUBIX database. Either the *rubix-dev* policy must be altered to allow unlabeled connections (not recommended) or the connection must be labeled using *netlabel* or *IPSec*.

As a simple example, the *netlabelctl* command (*man netlabelctl* for more information) may be used to statically assign an SELinux context to unlabeled communications based upon the IP address of the remote host. Note that the *netlabel_tools* package may need to be first installed on your platform. The following commands (as the *root* user) will assign different contexts to the unlabeled network traffic depending on if it comes from host 192.168.1.29 or 192.168.1.28. Note that the `\` character is not part of the command arguments and is meant to indicate that the following line should be joined with the current line.

```
netlabelctl unlbl add default address:192.168.1.29 \  
label:rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s0
```

```
netlabelctl unlbl add default address:192.168.1.28 \  
label: rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s1
```

Using the *netlabelctl* command will apply the mappings only during the current operating system session. A reboot will remove the mappings. To make *netlabel* mappings permanent place the *netlabelctl* command **arguments** into the */etc/netlabel.rules* file. For the preceding example, the *netlabel.rules* file would contain the following two lines:

```
unlbl add default address:192.168.1.29 \  
label: rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s0
```

```
unlbl add default address:192.168.1.28 \  
label: rxdev_u:rubix_remote_client_r:rubix_remote_client_t:s1
```

The *netlabel* service must be explicitly enabled. The *netlabel* service may be enabled by using the *system-config-services* program. The program may be started directly from the command line or through the *System->Administration->Services* menu. Once the program has started and initialized, the *netlabel* service may be enabled by scrolling down to the *netlabel* entry, selecting it with the mouse, and then choosing *Enable* at the top-left of the program. The *netlabel* service will now automatically start with every system startup. To start it immediately choose *Start* at the top of the program.

Uninstalling Trusted RUBIX

To uninstall Trusted RUBIX erase the installed packages and then recursively remove the */var/lib/RUBIXdbms* directory. To erase the Trusted RUBIX packages become the *root* user and either assume the *sysadm_r* role or place the SELinux policy into permissive mode and issue the following commands in the given order:

```
rpm -e rubixdbms-doc  
rpm -e rubixdbms-devel  
rpm -e rubixdbms  
rpm -e rubixdbms-odbc
```

Execution and Use

Assuming a Trusted RUBIX Role

Roles may be assumed by logging in as a user with the desired role configured to be the user's default role or by using the *newrole* command. If the *newrole* command is not available you may need to install the *policycoreutils-newrole* package.

To use the *newrole* command to assume a new role there must be SELinux policy rules that allow transition from the source role to the target role. By default, Trusted RUBIX allows the administrative roles to be reached **only** from the *staff_r* role and the client roles to be reached from the *user_r* or *unconfined_r* roles. The *unconfined_r* is only available using the Targeted policy. In order to assume the *staff_r* role you will need to create or configure a user to use that role. One way to do this is to create a Linux user and map that Linux user to the SELinux *staff_u* user. Then, the desired Trusted RUBIX roles may be added to the *staff_u* user using the SELinux configuration tool.

As an example, to reach the *rubix_dbadm_r* role first become the *staff_r* role. This is usually accomplished by having the *staff_r* be the default role of a logon user. Then the *newrole* command is used to assume the new role as follows:

```
newrole -r rubix_dbadm_r -t rubix_dbadm_t
```

Additionally, the *newrole* command may be used to change the current type or MLS/MCS level.

The terminal type must be defined as being a "secure" terminal type. If this is not already configured, it may be accomplished using the following steps:

1. Determine the terminal type by issuing the following command:

```
ls -Z `tty`
```

2. Add the type to the */etc/selinux/targeted/contexts/securetty_types* file (if you are using the Targeted policy) or the */etc/selinux/mls/contexts/securetty_types* file if you are using the MLS policy.

Starting the Trusted RUBIX Dispatcher

Prior to initiating database sessions the dispatcher service must be started. This is done using the *rxsvrman* command. Note that the command must be performed by a user currently operating in the *rubix_op_r* or *rubix_dbadm_r* role.

Example:

```
rxsvrman -s
```

The dispatcher and all instantiated, idle servers may be terminated as follows:

```
rxsvrman -t
```

Creating a Database

Prior to performing SQL operations a database must be created. This may be done using the *rxisql* command. The *rxisql* client is a text based command tool used to submit SQL operations to the server software. Note that the command must be performed by a user currently operating in the *rubix_dbadm_r* role. The database may be created as follows (the examples assume a local connection):

```
rxisql -d master
rxsql> create database MyDB;
rxisql> q
```

Note that the *master* database is used to create other databases. The *master* database need not be created.

The database may be dropped using the *rxdb* command as follows:

```
rxdb -d MyDB
```

Performing Client SQL Operations

The *rxisql* client is a text based command tools used to submit SQL operations to the server software. Note that the command must be performed by a user currently operating in the *rubix_client_r* role or one of the administrative roles. Note that the database connection string (argument to the *-d* option) contains the database name only (e.g., *MyDB*) for local connections and a remote host and optional port number for remote connections (e.g., *MyDB@my.host.com:4156*) The *rxisql* command may be used to perform SQL operations as follows (the examples assume a local connection):

```
rxisql -d MyDB
rxsql> create table MyTable (col1 int);
rxsql> insert into MyTable values 555;
rxsql> commit;
rxsql> q
```

Configuring and Creating ODBC Applications

ODBC applications are custom programs that may access standards compliant DBMS's. ODBC is Microsoft's adaptation of the X Open CLI standard. The following website gives details of its use, including the API specification:

[http://msdn.microsoft.com/en-us/library/ms710252\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms710252(VS.85).aspx)

Please refer to the Trusted RUBIX ODBC Guide for details about building applications with ODBC and configuring the Trusted RUBIX ODBC client to interoperate with pre-existing ODBC compliant applications.

Overview of User Documentation

All user documentation is in PDF format and is found in the `/var/lib/RUBIXdbms/doc` directory. The following is a brief overview of the documentation.

- ***RX_SELinux_Guide.pdf***
The Trusted RUBIX SELinux Guide describes the SELinux security mechanism as it relates to Trusted RUBIX and how to create custom SELinux DBMS policies.
- ***RX_Trusted_Facility_Manual.pdf***
The Trusted Facility Manual describes administrative aspects of the DAC and MAC security policies.
- ***RX_Security_Features_Users_Guide.pdf***
The Security Features User's Guide describes aspects of the DAC and MAC security policies as they relate to normal database user operations.
- ***RX_Commands.pdf***
The Trusted RUBIX Commands Reference Guide contains instructions on the proper use of the administrative commands and their security behaviors.
- ***RX_SQL_Guide.pdf***
The SQL Reference Guide contains the details on the use of the SQL language specification.
- ***RX_ODBC_Guide.pdf***
The ODBC Reference Guide contains the details of the ODBC programming specification, application build instructions, and configuration information.
- ***RX_Information_Schema.pdf***
The Information Schema Guide describes all schema views and their security behavior.
- ***RX_Security_Policy_Manager_Reference_Guide.pdf***
The Security Policy Manager (SPM) Reference Guide describes the architecture, use, and the Security Policy Markup Language (SPML) of the Security Policy Manager. The Security Policy Manager implements an Attribute Based Access Control (ABAC) policy using an XML language that is based on X Open's XACML standard.
- ***RX_Security_Policy_Manager_Tutorial.pdf***
The Security Policy Manager Tutorial gives step-by-step examples of building and using the SPML language to create a variety of tailored security policies.
- ***RX_SQL_Tutorial.pdf***
The SQL Tutorial gives step-by-step examples to using Structure Query Language (SQL) and building ODBC applications.

Support and Further Information

The most recent versions of the documentation are posted at <http://www.rubix.com/> for download. Support and further information may be acquired by email at support@rubix.com.