Administration of the Annual Reporting System

RFPK uses software known as the *Automated NCRR Annual Report Package* for preparing its annual report for the NIH. This document describes the installation and maintenance of this package.

The information in this document is targeted primarily toward the RFPK Software Team and associates and is specific to the computer systems and network installed in the RFPK Laboratory of the Department of Bioengineering of the University of Washington. RFPK is the Resource for Population Kinetics. Its work is supported, in part, by grant P41 EB-001975 of the National Institutes of Health (NIH) of the U.S. Department of Health and Human Services.

Table of Contents

Introduction

RFPK uses software known as the *Automated NCRR Annual Report Package* for preparing its annual report for the NIH. This software was obtained from Michael Riffle, of the Department of Biochemistry, who may have been its author. The version of the package that we have seems to have been developed or configured for the Yeast Resource Center (YRC). The acronym *YRC* appears as the database name and in several path names of the configuration. In this document the software will generally be referred to simply as the *package*.

Security

The package is a web application built around a MySQL database. It appears to have been designed so that various researchers could use it to maintain information about their own projects. Because they would interact with the package via the public internet, a security scheme was implemented, which included the concept of session and the use of the secure socket layer (SSL). With the original security architecture intact, it would have required a fair amount of work to port the system to current versions of the Apache web server and the MySQL relational database management system. Since Paolo Vicini is the only user of the package at RFPK, it was decided to disable the SSL features. The security issue, nevertheless, had to be addressed, due to the fact that Paolo's workstation resides on the public internet. The solution chosen was to rely on the firewall to bar access to all IP addresses except that of Paolo's workstation.

The domain name of the firewall is `spk.rfpk.washington.edu`. The package can be securely accessed from outside the firewall by using two of the firewall's features:

- Port forwarding. Traffic to an IP port on `spk.rfpk.washington.edu` can be forwarded to the same or any other port on any host behind the firewall. A port number must be chosen that is not already being forwarded to another machine. For this package we use the port 8082 and have it forwarded to port 80 on the server on which the package is installed.

- External access control. A list of IP addresses that can access this port must be provided. This list would include the IP address of Paolo Vicini's workstation. If required, additional IP addresses could be added.

Installation

The package, including the web server part of the application, is currently installed on the RFPK database server machine. This section assumes that its is desired to install the entire package on another server or to reinstall it on the current machine.

Creating a Database and an Administrative User

It is assumed that MySQL is already installed on the machine on which you will install the package.

At the command line prompt, issue the command

```
mysql -uroot -p
```

and enter the database root password when requested.

At the MySQL command prompt enter the following SQL statements:

```
use mysql;
create database YRC;
grant all on YRC.* to admin@localhost identified by '4SPKdb%';
quit;
```

Populating the Database

It is assumed that you have available a database backup called `YRC.sql`, previously created using the *mysqldump* program. This file will now be used to populate the database just created.

At the command prompt, input the following command

```
mysql -uadmin -p YRC < YRC.sql
```

and enter the administrative password when requested.

Installing General Perl Modules

The package, written in perl, depends on certain perl modules.

- Digest::MD5
- RTF::Writer
- DBD::mysql

They may already be installed on the server. If not, obtain them from CPAN.

One way to determine if a module is present is to discover whether or not its man pages have been installed. For example, the command

```
man RTF::Writer
```

will output the manual page for RTF::Writer to the screen, if that module is installed.

Installing the Package's Own Perl Modules

The packages own perl modules are located in the `r2` repository, starting at the directory `r2/src/apps/annual_report/YRC`. To install the modules, they most be made executable and copied to the `site_perl` library. Here is the procedure to follow:

1. If it does not already exist, get a CVS workspace in your home directory on the server. The procedure for doing this is described in the CVS RFPK HOWTO [1].
2. Change directory to `r2/src/apps/annual_report/YRC`.
3. Use the *install* script that is located in the directory to complete the installation:

```
chmod +x install
su
./install
exit
```

Installing the Static Web Content

The static web content of the package consists of `.html` files, images and stylesheets. These files are located in the `r2` repository in the directory `r2/src/apps/annual_report/HTML`. They must be copied to the web server content directory. Here is the procedure to follow:

1. Change directory to `r2/src/apps/annual_report/HTML`.
2. Use the *install* script that is located in the directory to complete the installation:

```
chmod +x install
su
./install
exit
```

Installing the CGI Scripts

The package uses CGI scripts to create web pages on dynamically. These files are located in the `r2` repository in the directory `r2/src/apps/annual_report/CGI`. They must be copied to the web server `cgi-bin` directory. Here is the procedure to follow:

1. Change directory to `r2/src/apps/annual_report/CGI`.
2. Use the *install* script that is located in the directory to complete the installation:

```
chmod +x install
su
./install
exit
```

Verifying the Installation

You should now be able to run the package.

If the server on which you have installed the package has a graphical desktop installed, log in to the desktop and start the browser. Then go to the following URL:

`http://localhost/cgi-bin/YRC/projects/searchProjects.cgi`

If the server does not have a graphical desktop, you should be able to access it from another computer. If, for example, the server is behind the firewall and has an IP address of `192.168.1.8`, the URL would be:

http://192.168.1.8/cgi-bin/YRC/projects/searchProjects.cgi

Configuring the Firewall

As explained above, you need to configure the firewall to forward traffic directed to port 8082 on the firewall to port 80 on the report server. You should also attach to this forwarding action an access list including, at least, the IP address of Paolo Vicini's workstation. When that is done, verify that it works by accessing the application from Paolo's workstation using the the following URL:

http://spk.rfpk.washington.edu:8082/cgi-bin/YRC/projects/searchProjects.cgi

Backup

Backup Strategy

At present, the package is installed entirely on whitechuck, which is backed up to tape on a weekly basis. This is a *raw* backup, in that everything is simply copied to the tape. Much of the information is in a binary form. This information is certainly sufficient to restore the database in case of damage. In order to accomplish this, all of the files associated with MySQL could be copied from tape to disk, using the *restore* function of the *xbru* utility, which is part of the tape backup system installed on whitechuck.

Restoring from the backup tape should only be used as a last resort, however, for the following reasons:

- The tape backup is taken only once a week. Information added to the database between the time of the last backup and the time of the database failure is lost.
- Care must be taken to restore only the files associated with the YRC database and not with the SPK databases. This is a risky undertaking. If not done properly, transactions might be lost from the SPK databases.

A nightly symbolic backup is also taken. A script that uses *mysqldump* is used. It is run on a workstation in the home directory of the workstation's principal user (currently jambutty, used by alan). This way, the dump goes onto a hard disk on a machine separate from the server. The user's home directory is also backed up to whitechuck each work day night, providing another copy. On whitechuck, these home directory backups are retained for a certain period, providing the ability to revert the database several days, if needed.

Configuring Backup

This section describes the way that nightly backup was configured on jambutty. The same procedure could be followed to backup to another workstation.

1. *Install a read-only database user.* At the command prompt, start *mysql* as the root database user by issuing the following command: **mysql -hdbserver -uroot -p**. Then enter these SQL commands:

```
use mysql;
grant select on YRC.* to reader@"%" identified by "reader";
grant lock tables on YRC.* to reader@"%" identified by "reader";
quit;
```

2. *Install a backup script.* Using your favorite editor, create the following script called `~/bin/shell/backup-YRC`

```
#!/bin/sh

cd $HOME
mysqldump -ureader -preader -hdbserver --opt YRC > YRC-backup

exit 0;
```

   Make the script executable by enter the command **chmod +x ~/bin/shell/backup-YRC** at the command prompt.

3. *Add a command to your private crontab.* At the command prompt, issue the command **crontab -e**. This will bring up a text editor window showing the current crontab. Add the following line:

```
20 2 * * 0-6 ~/bin/shell/backup-YRC
```

   then save the file and exit.

Copyright Notice

Notes

1. ../cvs/cvs.html
2. http://www.opencontent.org/openpub/