**PET Center IT Management**

1. User management

Yale PET Center uses Yale netid as user account. Each user will get an account with his/her netid as username. Depending on user’s role, user can have different privileges, which contain different privileges in user groups, database, webapp, etc.

Yale PET Center uses ITS central directory server to authenticate users. Each normal user will ask for netid and associated password to access PET center resource. PET Center never record, store or change the password.

For some special reason, we also create local account with local password. This is only for user without a valid netid. Users in this category are possibly local pseudo user, external co-operators who want to transfer data, previous PET center people who still need access to finish some left-over works, etc. In this case, both username and password are stored in PET center server (in encrypted format).

We review user account annually. This is done by a program which can find any account which doesn’t login for 6 months. These accounts are reviewed manually and disabled from access.

All user accounts and their associated privileges can be found here: <http://orionpet.med.yale.edu/IT/information/pet%20user.htm>

2. Software management

Yale PET Center has 13 Linux servers/workstations. Each has many open-source and commercial software installed. Besides these, we have a 56-node cluster with special software installed. Major software installed on workstation can be found here: <http://orionpet.med.yale.edu/IT/information/software.htm>

Yale PET Center also has self-developed software to fit our needs. These software are developed in different languages and tools, and are categorized into several projects. All source codes are managed by version-control system, such as Subversion or Git. All changes in these source codes are tracked by the version-control system. For details, please see this: <http://orionpet.med.yale.edu/IT/information/Source%20codes.htm>

3. Network security

Yale PET Center has its own firewall which protects its resource from attacking and unauthorized access. The firewall is administrated by Yale ITS. All IT resource are inside firewall and cannot be access from outside unless opened ports in firewall. For a list of opened ports, please see this: <http://orionpet.med.yale.edu/IT/information/firewall.htm>

Yale PET Center has its own VPN by which user can access PET center IT resource. The VPN is part of Yale University VPN but with different profile. It is administrated by Yale ITS. Only a few authorized users are in the VPN user list. The PET VPN is mainly for users who work remotely, such as home or travel. It’s unnecessary inside PET Center.

We have a special server dracopet.med.yale.edu, which allows some user access by SSH. This is mainly for user who doesn’t have valid Yale netid and cannot use PET VPN. This server has internal iptable-based firewall, which allows user access only from specific IPs. This firewall is administrated by PET center.

4. Backup and archive

Daily backup: this contain user data(home1,home3), source codes, documents, database, webpage, log files, etc (data1), Dicom image files (data8), and server files (orionpet,tauruspet,hydrapet,leopet, lynxpet, geminpet, librapet). The schedule is quarterly per year. On the first day of each 3 months, we do a full backup. Then on every night else, we do an incremental backup (i.e. copy all new or changed files since last backup). We try to keep the backup tapes for 6 months before they are overwritten.

All raw data from five scanners will be transferred to SAN when scans are done. On every morning at 5:00am, we copy the raw data(data3,data4,data5) to tape for archive. This is a daily operation. To avoid missing something, we do a yearly full archive at Dec 24. These tapes are stored in LMP108A. We also do a monthly full archive of Dicom files. This happens on the fourth Saturday of each month. These tapes are stored in PET building first floor.

For raw data archive tapes, we will make an extra copy every week. The cloned tapes will be stored in PET building first floor.

Detailed schedule can be found here: <http://orionpet.med.yale.edu/IT/information/backup_schedule.htm>

5. Haven Database

Haven is a MySQL-based database, and contains many information related to our daily works. Inside, information are stored into several (about 20) tables. Each table contains many columns, and each record is a row in the table. Different tables are linked with each other by foreign keys, which is so-called relationship. Most tables contain keys and indexes, which prevent duplication and make search more quickly.

Among the 20 tables in haven database, 5 tables are more important. They are Patient, Visit, Injection, Study, Series. For these 5 tables, each record has its own CreateDateTime and ModifyDateTime so that we can track when the record was created and modified.

We have many programs to communicate with haven database and query/update database information. These programs usually hide database technical details for users so that they can use with knowing MySQL. We also have a webpage for users to access haven database. In this webpage, users are granted to different privilege so that they can access/modify different contents in haven database according to their privilege.

Haven database is backed-up every hour during working hours, and all backed to tape every night.

6. ePHI

Most data in PET Center system are anonymous. Only the Patient table in haven database contains patient real name and birthdate. We treat them very carefully by limiting user access these information.

Only very few users are granted privilege to access these information. All other users are not allowed to access in any method, such as webpage, programs, etc. For users who can access these information, their computers should be encrypted by PGP or BitLocker.

There are a few scanned hand-writing document which may contain patient information. They are saved as Analysis. The database label them as ePHI data type and allow only authorized users to access.

7.