www.project-open.com

]project-open[V3.1

Advanced Unix Maintenance Guide

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

Frank Bergmann,
V1.11, 2006-06-10

INDEX

1 About this Document

1.1 Version

Version: 1.11, 2006-06-10

Author: Frank Bergmann

Status: Advanced Draft

Path: H:\Tech\Infrastructure\Hosting\Marktetplace System Operations & Security Guide yymmdd.doc

1.2 Scope

This manual describes the installation and operation of the OpenACS 3.4.8 TCL toolkit on a Linux server with SuSE 6.4 or Red Hat 6.0.

1.3 Audience

The manual is written for Unix system administrators. However, a considerable part of the manual should also be understandable for less technical persons.

1.4 Version History

- V1.1 -> V1.2: 040510 Frank Bergmann: Update
- V1.0 -> V1.1: 040215 Frank Bergmann: Integration of several text fragments from different sources

1.5 ToDo's

- Explain the Apache setup, including the "excuse" screen and BigBrother.
- Protect BigBrother directory with password.
- Check for tmp symlink attack at the Oracle backup script.
- Update the firewall policies and rules to reflect recent (010510) changes.
- Start Oracle 8i listener before importing a database dump.
- AOLServer SSL Certificate configuration.

2 System Overview

2.1 Context Overview

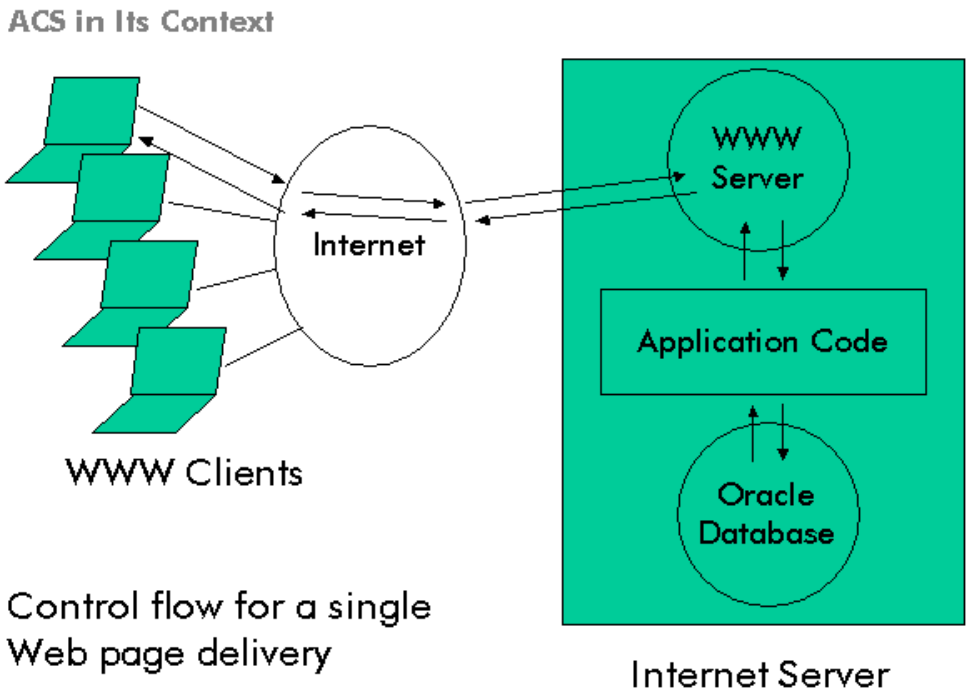

Figure 2-1

An OpenACS server in general consists of the following elements:

- a web server (AOL Server),
- the OpenACS application code and
- the Oracle 8i database

A Unix/Linux server hosts these processes.

2.2 Operation Overview

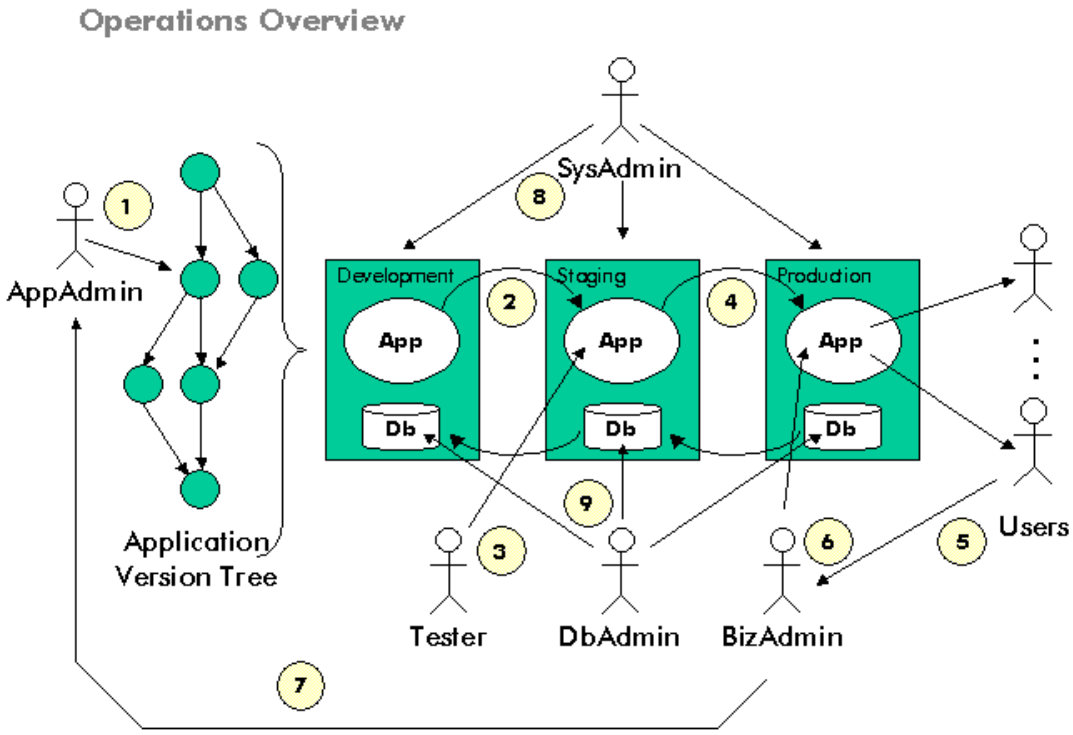

Figure 2-2

The image above gives a brief overview about the processes involved in updating a productive OpenACS application:

1. The **Application Administrator (AppAdmin)** implements a new feature or fixes a bug using the **Development Server**.
2. The System Administrator (SysAdmin) transfers the application to the Staging Server.
3. A dedicated **Tester** assures the Quality Control at the **Staging Server**.
4. The **System Administrator** transfers the application to the **Productive Server**.
5. **Users** are using the application. Suggestions and bugs are reported to the **Business Administrator (BizAdmin)**.
6. The **BizAdmin** maintains the application data: He adds new products and he opens up new transactions.
7. The **BizAdmin** reports bugs and suggestions to the **AppAdmins**

8. The **SysAdmin** keeps the servers running and maintains and supervises the staging procedure.
9. The **DbAdmin** keeps the databases running and supervises Db backups.The operation of a “productive” OpenACS has to deal with a range of critical situations such as:

- Mistakes of system administrators or software developers
- Failures of both hard- and software
- Software updates of the online application
- Exhaustion of resources such as hard disk space, RAM and processing power

The goal of the operation is to maintain high system availability. An efficient organization in combination with good network infrastructure and reasonable hardware can reach a level of 99.8%.

2.3 Roles and Responsibilities

Role	Responsibility
SysAdmin	Keeps the servers running
DbAdmin	Keeps databases running
AppAdmin	Administrates the application running and fixes application level errors
BizAdmin	Understands the client’s business context and provides the information how to apply the application.

Table 2-1

3 Backup

3.1 Concept, Tradeoffs and Decisions

The backup and recovery concept is structured in three stages to construct the working application:

1. Operating System (Hardware, Solaris, SSH, …)
2. Applications (Oracle and AOLServer) and
3. Data and code

Both, the backup and recovery processes are oriented around these three basic steps and the alternative pathes that lead to a successful recovery.

	Data	Backup	Restore
Operating System	2nd Disk backup	Copy the entire disk	Swap disks
	Tape backup	Copy the OS to tape	Restore tape from running minimal installation
	Solaris install from scratch	Original installation media	Install
Applications	Oracle Tar	Tar the /ora directory to tape or disk	Restore the Tar
	Oracle install from scratch	Original installation media	Install
	AOLServer Tar	Tar the /usr/local/aol* to tape or disk	Restore the Tar
	AOLServer from scratch	Original installation media	Install
Data/ Code	Oracle Dump	Write Oracle dump	Restore Oracle dump
	Oracle Full Backup	Tar the /ora directory to tape or disk	Restore the Tar
	Application Code	Tar the /web directory to tape or disk	Restore the Tar

The main challenge in this environment is to make sure that a specific version of the application code is working together with a specific data model.

3.1.1 Tradeoffs and Decisions

Database backup is a standard process covered in detail by the Oracle Administrators Manual. Various options exist.

- Oracle Hot Backup:
  The productive server is backed up using a “dump” database export that saves the content of the entire database into a flat file. This file is moved to a backup location.
  This process has the effect of slowing down the database operations, but

does not affect the availability, so that several daily backups are possible, preferably during low traffic hours (morning, lunch, night).

- Full Backup:
  In addition, the productive server is backed up once a week fully.

The reason to choose this procedure is that the size of the productive database is relatively small (compressed less then 100mByte) and that we want to backup the DB several times a day. So an online backup is necessary, and the “dump” procedure is much easier to handle than a backup based on redo logs.

To avoid security holes, all database dumps are encrypted using PGP at the productive server and are stored in encrypted form at the backup server.

3.2 Full Backup

A full backup needs to be made every week with the predominant spare disk configuration, copying the entire hard disk to the spare disk. This procedure includes the shutdown of the Oracle database because it includes a normal shutdown of the computer.

An excerpt from the root crontab:

```
37 1 * * 7 /root/.backup/pre_backup.pl
```

3.3 Application Backup

Both Oracle and AOLServer are slowly changing data, being updated every few months. Both application are updated every few months from installation media.

3.3.1 Oracle 8i

Tar the content of the /opt/oracle directory to a tape or a CD. In addition, the following files have to be backed up:

```
/etc/oratab
/etc/oraInst.loc
/etc/profile/oracle/*
```

3.3.2 AOLServer

The AOLServer installation has similar characteristics as Oracle. The following files have to be backed up for a full recovery:

```
/usr/local/aol*
/sbin/services/*
```

3.4 Database Backup

The backup implementation is based on two scripts:

- One script (“export-oracle”) creates the backup “dump” and sends it to a backup server via Email.
- The other script (“dump_backup.perl”) receives the backup “dump” from Email and stores it into a specific directory.

An entry in the root crontab at the productive server starts the “export-oracle” script:

```
10 */3 * * * /usr/sbin/export-oracle 2>&1 | tee /var/log/oraback/oraback.`/bin/date
+\%Y\%m\%d.\%H\%M`.log | egrep -i "warning|error|fatal" | grep -iv "without warnings"
```

An entry in the root crontab at the receiving server (Barna) deletes entries that are more then 7 days old:

```
20 1 * * *      find /mnt/megaraid3/cluster/Tech/dump_backup -ctime +7 -name '*dmp.gz*' -exec rm {} \; >>
/var/log/temporary_internet_files.log 2>&1
```

3.4.1 Export-oracle

This export script is called by the “crontab” of the productive server:

```
[root@www1 /root]# cat /usr/sbin/export-oracle
#
# /usr/sbin/export-oracle
# V1.1, 010412 Frank Bergmann <frank.bergmann@project-open.com>
#
# V1.0 -> V1.1:
#       - Now using COMPUTER_NAME as first part of dump
#       - Now encrypting using PGP
#       - Now sending out to centralized backup server
#

HOME=/home/oracle
HZ=
LOGNAME=oracle
ORACLE_BASE=/opt/oracle/app/oracle
ORACLE_HOME=$ORACLE_BASE/product/8.1.6
PATH=$PATH:$ORACLE_HOME/bin
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
ORA_OWNER=oracle
ORACLE_SID=ora8
ORACLE_TERM=vt100
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
PATH=$ORACLE_HOME/bin:$ORACLE_HOME/lib:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/s
bin
SHELL=/bin/sh
TERM=vt100
TZ=US/Eastern
CURRENT_TIME=`/bin/date +\%Y\%m\%d.\%H\%M`

# Change these!
COMPUTER_NAME=<computer_name>
SERVICE_NAME=<service_name>
DATABASE_PASSWORD=<database_password>
BACKUP_EMAIL=<backup_user_email>

exportdir=/var/log/oraback
file_body=$COMPUTER_NAME.$SERVICE_NAME.$CURRENT_TIME.dmp
file=$exportdir/$file_body

echo "exporting to $file"
su - $ORA_OWNER --command="exp $SERVICE_NAME/$DATABASE_PASSWORD file=$file owner=$SERVICE_NAME
consistent=Y"

echo "zipping and encrypting"
gzip $file
pgp -e $file.gz $BACKUP_EMAIL

# archieve dump at backup server
uuencode $file.gz.pgp $file_body.gz.pgp | mail $BACKUP_EMAIL
```

The following script is called by the QMail mail system working at the backup server whenever a mail arrives for the backup_user. Check that it has write permissions in the destination directory:

```
# cat ~backup_user/.qmail
| ~backup_user/bin/dump_backup.perl >> /var/log/dump_backup/dump.`/bin/date +"%Y%m%d"`.log 2>&1
```

When working with sendmail, the script has to be named “.forward”.

3.4.2 dump_backup.perl

The following script is called by the QMail mail system working at the backup server whenever a mail arrives for the backup user (be sure to create ~backup_user/data directory):

```
#!/usr/bin/perl
#
# dump_backup.perl
#
# 010412 Frank Bergmann <frank.bergmann@project-open.com>
# Receives incoming backup file by qmail/sendmail,
# extracts header information, stores them temporarily
# in the data directory and calls uudecode to restore
# them.

$debug = 1;
$home_dir = "/home/backup";
$data_dir = $home_dir."/data";

$now = `/bin/date +"%Y%m%d.%H%M"`; chomp($now);
print "$now: dump_backup.perl: receiving file\n" if ($debug);

# ignore the header information and start writing the
# file with the begin of the UUEncoded data:
#
$file = $data_dir."/incoming.".$now;
open(F,"> $file");
open(G,"> $file.all");
$flag=0;
while ($line = <STDIN>) {
    print G "$line";

    if ($line =~ /^From/) { print "$now: dump_backup.perl: $line"; }
    if ($line =~ /^begin /) {
        $flag = 1;
        print "$now: dump_backup.perl: $line";
    }
    if ($flag) {
        print F "$line";
    }
}
close(F);
close(G);

# uudecode the file
# ToDo: Check for tmp symbolic link attack!!!
$return = system("cd $data_dir; /usr/bin/uudecode $file > /tmp/uuencode.log 2>&1");

$now = `/bin/date +"%Y%m%d.%H%M"`; chomp($now);

if ($return == 0) {
    system("rm -f $file");
    system("rm -f /tmp/uuencode.log");
    print "$now: dump_backup.perl: finished\n" if ($debug);
} else {
    $err_msg = `cat /tmp/uuencode.log`;
    print "$now: dump_backup.perl: error: $err_msg \n";
    system("rm -f /tmp/uuencode.log");
}
```

3.4.3 PGP Configuration

Both sides of the channel use a standard PGP installation to encrypt the database dump. You need to create a PGP public/private key for backup_user@your_company.com. The public key of the backup server has been extracted using:

```
pgp -kx backup_user@your_company.com /tmp/backup.public.key.pgp
```

and imported into the public key ring of the productive server using:

```
pgp -ka /tmp/backup.public.key.pgp
```

4 Failure Recovery

The purpose of this chapter is to describe how to recover a Project/Open application after any kind of incident, ranging from administration mistakes to a complete hardware failure.

4.1 Concept, Tradeoffs and Decisions

The recovery of the system proceeds in 3 major steps that are reflected by the following subchapters:

4.1.1 Assumptions

We assume about the productive server environment:

- That a skilled system administrator is available 24/7 to
  1. Manage the situation,
  2. Communicate with the hosting support and
  3. Perform the necessary SysAdmin recovery steps
- That the hosting support reacts within about an hour to critical situation.
- That the productive server lives in a state of the art hosting center with reasonable infrastructure availability (network & power supplies)
- That the productive server hardware can be replaced in case of a complete failure

4.1.2 Challenges

The main problem with recovery is to avoid a reinstallation of Oracle8i and the AOLServer. In particular Oracle8i causes a considerable delay of 5-10 hours if the 500Mbyte pre-configured TAR has to be transferred through a 128kBit Internet connection.

On the other hand application code and the database content can be restored much faster because:

- The application code is a few MBytes compressed and is maintained by Project/Open, so that it can be regenerated at any time.

- The database content is 3-5Mbyte compressed and is backed up every few hours and sent to the backup server (Barna).

Thus the Oracle installation and the Oracle database content are the critical elements in this backup concept

4.2 Failure Types

All failures that can occur in the productive can be classified into one of the following categories:

1. Primary Hard Disk Failure:
   The main hard disk either fails or a system administrator's failure deletes its content. The secondary hard disk survives.
2. Hardware Failure of both disks (burning server):
   Both hard disks get damaged, for example by a fire or after an unfriendly intrusion.
3. Non-Disk Hardware Failure:
   Some hardware component of the productive server fails such as the power supply or a network card.
4. Hosting Infrastructure or Power Failure:
   Something at the hosting site goes wrong which inhibits the access to the productive server such as power or network shortages.

4.2.1 Recovery Procedures

To deal with the above types of failures, the following procedures are available:

Hosting Center:

1. Swapping the server hard disks:
   The Hosting support can swap the server hard disks on request.
2. Hardware repair:
   The Hosting support can repair defect hardware or replace the entire server by backup hardware within 24 hours.
3. Providing a completely new server hardware

System Administration:

4. Update the Oracle DB data
5. Update the application TCL code
6. Reinstall the Oracle database
7. Reinstall the AOLServer and OpenACS system

The division point between the two companies is that Hosting is responsible to provide a working (Linux) operating system while the System Administrator is responsible to bring the application up and running.

4.2.2 Failure Types Against Recovery Procedures

The following table details the steps to be taken in case of a failure:

Failure Type	Recovery Procedures
Primary Hard Disk Failure	1, 4, 5: Fallback to the backup disk and recovery of the latest application data
Hardware Failure of both disks	2, 6,7,4,5: Complete reinstall of the server
Non-Disk Hardware Failure	2: Repair of the server hardware, maintaining the original disks and their content.
Hosting Infrastructure Failure	None: We just have to wait until Hosting gets the servers up again. We might check for an alternative provider in case of a longer failure.

Table 4-1

4.2.3 Recovery Timing

Please see the attached Gant chart for a timing of the recovery procedures. This chart assumes a Primary Hard Disk Failure:

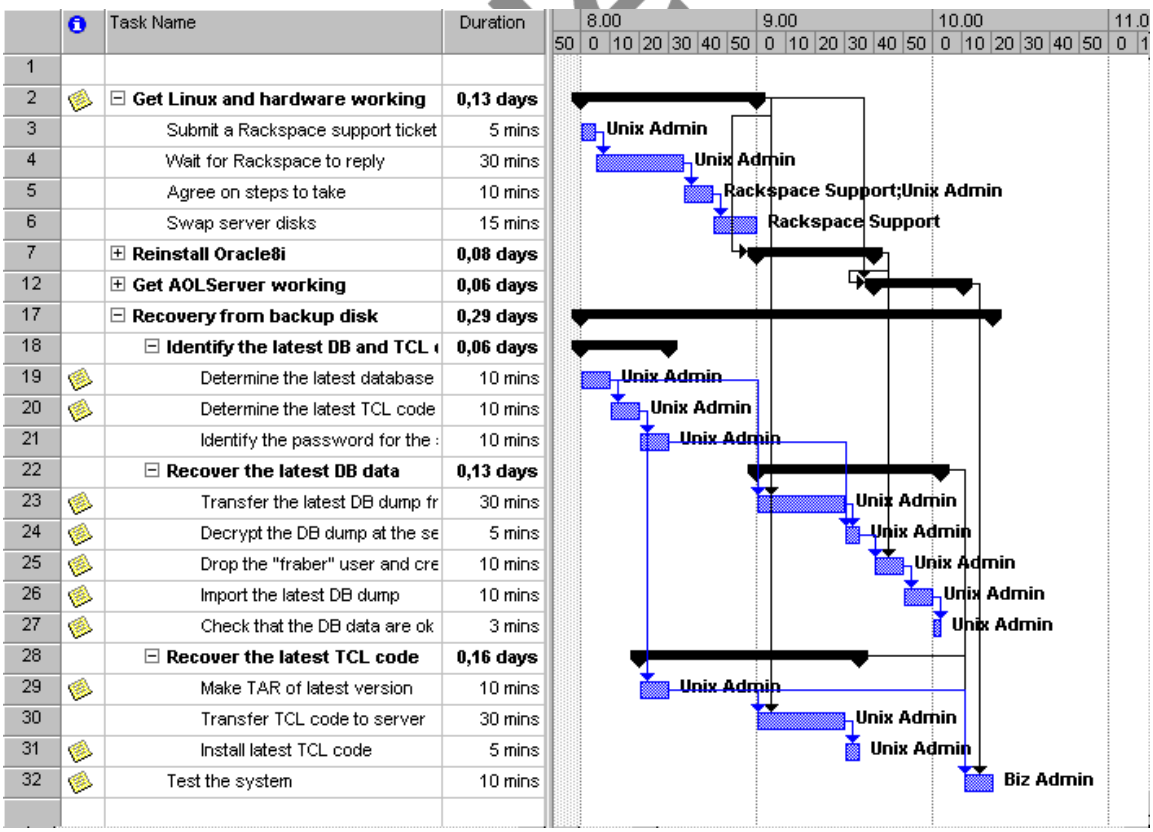

Figure 4-1

4.3 Linux Operating System Recovery

There procedures deal with the restoration of the productive server to the point where the Linux operating system is working again

4.3.1 Swapping the server hard disks

On request, Hosting can perform a swap of the server hard disks within 1-2 hours, thus restoring the server state at the time of the last full backup. Please see the section “Full Backup” for details on how the backup is made.

4.3.2 Hardware repair

The Hosting support can repair defect hardware on request.

4.3.3 Providing a completely new server hardware

The Hosting support can replace the entire server by backup hardware within 24 hours.

4.4 Installation From Scratch

These procedures are used to setup Oracle, AOLServer and ACS after a complete crash up to the point where it is possible to upgrade to the latest backup of the application (database data + ACS code).

Please see the installation section for details.

4.5 Oracle and ACS Update Procedures

4.5.1 Recover the Current Application TCL Code

To recover the current version, extract the current version of the application from the CVS system at the development server: ToDo: Where to get the code?

1. Determine the current marketplace version of the productive server: In our server room we have a paper table at the wall with the complete release history. Check the latest release for the productive server.
2. Checkout the latest version from CVS:
   ```
   # cd web
   # cvs co -r version -d directory module
   ```
   with the parameters:
   version: the name of the version to recover (from paper sheet)
   directory: the target directory for the extracted code
   module: the name of the module (also from paper sheet)
3. Example:
   ```
   # cd web
   # cvs co -r eshkol-1-0-0 -d dev productive-acs
   ```
4. Tar the directory, transfer it to the productive server and untar it. That is all; the code is completely “passive” and fits to the version of the data model in Oracle.

4.5.2 Load the Latest Database Backup Dump into Oracle

The following procedure describes how to charge a running Oracle8i database system with the latest backup dump of the database. It assumes that the database is running and the user and the tablespace “intranet” exist.

1. Make sure that Oracle is up and the tablespace and the user exist:
   ```
   # sqlplus intranet/intranet
   SQL> select email from users;
   SQL> quit
   ```
   you should get a list of several emails.
2. Login onto Oracle as “system” and delete the user “intranet” (the default user of the Project/Open application):
   ```
   # sqlplus system/manager
   SQL> drop user intranet cascade;
   SQL> create user intranet identified by intranet default tablespace
   intranet temporary tablespace temp quota unlimited on intranet;
   SQL> grant connect, resource, ctxapp, javasyspriv, query rewrite to
   intranet;
   SQL> revoke unlimited tablespace from jruiz;
   SQL> alter user jruiz quota unlimited on jruiz
   SQL> quit
   # imp system/manager file=www1.intranet.yyyymmdd.hhmm.dmp
   log=/tmp/log.log
   ```
3. Repeat step 1.) to make sure the tablespace is OK and restart the AOLServer (“killall –9 nsd” or svc –t /sbin/service/intranet (when using daemontools). Pray that it´s working now (good hint from a catholic Irish system administrator who used to work here...).
4. Repeat step 1.) to make sure the tablespace is OK and restart the AOLServer (“pkill –9 nsd” or svc –t /sbin/service/intranet (when using daemontools). Pray that it´s working now. (Good hint from a catholic Irish system administrator who used to work with us once upon a time...).

4.6 Informing the Users

It is useful to inform the users about system maintenance when the Linux OS is working, but the ACS is still under maintenance:

4.6.1 “Not Available” Screen during ACS updates

If for any reason the ACS server or the database becomes unavailable, you have to swap the AOLServer (OpenACS) with the Apache web server (static pages).

Normally, AOLServer operates on port 80 (http) and Apache is installed to work on port 81. But during maintenance we take advantage of the preinstalled Apache screen to inform our users about the maintenance.

This is done by starting Apache on port 80 and the AOLServer on port 81:

1. Edit the Apache configuration file at /etc/httpd/httpd.conf and change the line “Port 81” to “Port 80”.

2. Edit the AOLServer configuration file at /home/aolserver/aolconf.tcl and change the line “ns_param Port 80” to “ns_param Port 81”
3. Stop the Apache server: “/etc/rc.d/apache stop” (SuSE 7.x) or “/etc/rc.d/httpd stop” (RedHat).
4. Restart the AOLServer: “killall –9 nsd” (it´s being restarted by Inittab or the daemontools).
5. Start Apache again: “/etc/rc.d/apache start” (SuSE 7.x) or “/etc/rc.d/httpd start” (RedHat).

5 Security

5.1 Concept, Tradeoffs and Decisions

5.1.1 Concept

Security in the context of our Marketplace application can be defined in terms of:

- Loss of Data:
  The chapter about backup deals with the strategy to avoid or limit the loss of data.
- **False creation or alteration of data:**
  This case has to be solved on the application level by introducing strong authentication measures because false creation of data can be due to company members, users or intruders.
  Currently (April 2001), there is no strong authentication build into the application because of user acceptance problems. So this is a political decision.
- **Distribution of Data:**
  Here application administrators, system administrators and intruders are the critical groups. The usual precautions are taken (see below).

5.1.2 Tradeoffs and Decisions

A major decision had to be taken concerning the security/accessibility of the productive database to application developers. Until now (April 2001) and probably during the rest of this year, a lot of changes have to be made to the application, including its data model, so an easy access of the developers to the productive database is essential. This means that the application data cannot be secured (neither access nor change) against application developers and system administrators.

To deal with this situation in a reasonable manner, we are currently (April 2001) introducing a “Bid Printer” to protocol all changes to the database on paper to make changes traceable.

Another decision has been made not to protect the productive server by an additional firewall (just the linux one). Instead, the “usual precautions” are taken to prevent intrusion by closing down al unnecessary Unix services and choosing strong passwords.

5.2 Implementation

5.2.1 Unix Level Security

The security concept on the Unix level is based on the idea to disable all unnecessary services and to analyzing the remaining services in detail for security. Following closely the “Bugtraq” security mailing list and updating the affected components guarantees the security of the Linux operating system itself.

Below please find a listing of the remaining services. Their respective security is further analyzed in the following chapters.

```
[root@www1 /root]# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:smtp                  *:*                     LISTEN  (Sendmail)
tcp        0      0 *:www                   *:*                     LISTEN  (OpenACS)
tcp        0      0 *:https                 *:*                     LISTEN  (OpenACS)
tcp        0      0 *:81                    *:*                     LISTEN  (Apache)
tcp        0      0 *:4829                  *:*                     LISTEN  (ToDo: Check process)
tcp        0      0 *:1984                  *:*                     LISTEN  (BigBrother)
tcp        0      0 *:ssh                   *:*                     LISTEN  (Secure Shell)
```

5.2.2 OpenACS Security

The OpenACS application is the most critical security risk in the system because it is under continuous development.

The overall OpenACS architecture consists of:

- **The AOLServer Web server**:
  The least critical component because it is an open source application being used by America Online (AOL) in their own environment. The development group is following closely the “AOLServer” mailing list to check for security exploits.
- **The OpenACS base system**:
  The ACS base system consists of a set of TCL pages that has to be checked for security together with the:
- **Application specific pages for our Marketplace application**:

The security check involves for each page the following criteria:

- Check that all pages require authentication for the appropriate user groups
- Check that all parameter that appear as part of a SQL query are formed appropriately.

Please see the separate document “Security Analysis Marketplace YYMMDD.xls” for a detailed analysis.

5.2.3 Apache Security

Some excerpts from /etc/httpd/conf/httpd.conf:

5.3 Configuration Details

```
# /etc/httpd/conf/httpd.conf
#
Port 81
DocumentRoot "/home/httpd/html"
User nobody
Group nobody
#
# Limit the load to Apache
#
MinSpareServers 2
MaxSpareServers 20
StartServers 2
MaxClients 50
```

CGI-Directory:

There is only one CGI directory with scripts from “Big Brother”:

```
# /etc/httpd/conf/httpd.conf
#
<Directory "/home/httpd/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

Content of /home/httpd/cgi-bin: These scripts form part of the current “Big Brother” distribution. Big Brother security exploits are reported to the monitored “Bugtraq” security mailing list.

```
bb-ack.sh
bb-hist.sh
bb-histlog.sh
bb-hostsvc.sh
bb-rep.sh
bb-replog.sh
```

5.3.1 SSH & Login Security

The regular telnet login is disabled.

Only the following users should figure in the /etc/passwd:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
www:x:100:101::/home/www:/bin/false
named:x:25:25:Named:/var/named:/bin/false
majordomo:x:91:91:Majordomo List Manager:/usr/lib/majordomo:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
squid:x:23:23::/var/spool/squid:/dev/null
fbergman:x:500:500::/home/fbergman:/bin/bash
rack:x:501:501::/home/rack:/bin/bash
oracle:x:504:504::/home/oracle:/bin/bash
nsadmin:x:505:508::/home/aol30:/bin/bash
webmaster:x:508:511::/home/webmaster:/bin/bash
bb:x:510:507::/usr/local/bb:/bin/sh
fax:x:10:14:Facsimile Agent:/var/spool/fax:/bin/bash
```

5.3.2 Sendmail Security

It is necessary to run Sendmail because the OpenACS application needs it to send Email. However, the access to the Sendmail port is blocked completely to the outside world.

```
# /etc/sysconfig/ipchains
#
:input ACCEPT
:forward ACCEPT
:output ACCEPT
-A input -s 209.61.155.151/255.255.255.255 -d 209.61.155.151/255.255.255.255 25:25 -p 6 -j ACCEPT
```

```
-A input -s 0.0.0.0/0.0.0.0 -d 209.61.155.151/255.255.255.255 25:25 -p 6 -j REJECT
```

5.3.3 Oracle Security

The Oracle database is not accessible from outside the computer. For this reason, the “lister8i” service has to be disabled.

6 Maintenance

6.1 Directories to watch for it's size

6.1.1 /var/log/aolserver

You can safely delete the "error" log files. They are used for debugging. The rest of log files record who has accessed that server via his browser. You can use them, for example, for statistical purposes, user profile, usage graphs or marketing campaign results.

6.1.2 Core files

The following command will find them:

```
Find / -name core
```

You can safely delete them.

6.1.3 Oracle Files

Check for the size used by Oracle itself (/opt/oracle) and it's databases (installed in /ora/):

/ora/m01

/ora/m02

/ora/m03

6.2 Big Brother

We suggest to use Big Brother for continuous monitoring of table spaces and disk partitions

6.3 Adding a new Virtual Server (ToDo: update)

Summary:

1. Adding the user to the Solaris system

```
# useradd -g nsadmin -m -s /usr/bin/bash <new_user>
# passwd <new_user>
```

2. Creating an Oracle tablespace

```
# su - oracle
# sqlplus system/manager
SQL> create tablespace <new_user> datafile '/ora/m02/oradata
/ora8/<new_user>01.dbf' size <tablespace_size>m autoextend on
default storage ( pctincrease 1);
SQL> create user <new_user> identified by <new_user> default
tablespace <new_user> temporary tablespace temp quota unlimited on
<new_user>;
SQL> grant connect, resource, ctxapp, javasyspriv, query rewrite to
<new_user>;
SQL> revoke unlimited tablespace from <new_user>
SQL> alter user <new_user> quota unlimited on <new_user>;
```

```
SQL> quit;
```

Check that it worked:

```
sqlplus <new_user>/<new_user>
SQL> select sysdate from dual;
```

A reasonable value for <tablespace_size> is “5”

3. Providing a copy of the ACS for the new user

```
# cp –r /web/<existing_acs> /web/<new_user>
# chown –R <new_user>:nsadmin /web/<new_user>
```

4. Loading acs into the user's new tablespace

```
# su - oracle
# cd /web/<new_user>/www/doc/sql
# sqlplus <new_user>/<new_user> < load-data-model.sql
```

This command takes several minutes. Please note that you might have to create module specific tables if the developers did not include them into the “load-data-model.sql” file.

5. Adding a new configuration file to /usr/local/aolserver

```
# cd /usr/local/aolserver
# cp aolconf.<org>.tcl aolconf.<new_user>.tcl
# vi aolconf.<new_user>.tcl
  -> Change “<org>” to “<new_user>”
```

6. Define an IP port in /usr/local/aolconf.config

```
# vi aolconf.config
-> Add a lines for “hostname”, “addresses” and “ports” for
<new_user>
```

7. Edit the /web/<new_user>/parameters/*.tcl config file:
   a. Globally replace “podemo” by “<new_user>

8. Edit /etc/inittab and add:

```
# ao:234:respawn:/usr/local/aolserver/bin/nsd-oracle -i -u
<new_user> -s <new_user> -t
/usr/local/aolserver/aolconf.<new_user>.tcl >/dev/msglog 2>&1
```

9. Check whether the server is up and running:
   a. Check for the process list

```
# ps –ef | grep nsd
```

This command should give you the list of running AOLServers (“nsd”). Make sure your new server is there “-s <new_user>”.

   b. Connect to the server using the values from aolconfig.config:

```
# telnet <ip_address> <port>
```

Type in (capital letters!)

```
GET /
```

You should get some HTML output consisting of an error message (because the “GET /” did not contail all necessary information).

c. Check for the server log file for notification of problems during startup. You may delete the file and restart the server if you find too many lines:

```
# less /var/log/aolserver/xdomen-error.log
```

6.4 Restoring Data from a TAR Backup


```
- Insert the right tape into the tape drive:
  With a tape changer you have to insert the
  tape in one slot of the magazin (example:
  slot 1) and then issue (the "#" indicates the
  start of the command line, it's not part of
  the command):

  # mtx load 1

  You can check that tape 1 is loaded correctly
  using.

  # mtx status

  >    Storage Changer /dev/changer:1 Drives, 6 Slots ( 0 Import/Export )
  >    Data Transfer Element 0:Full (Storage Element 1 Loaded)
  >    Storage Element 1:Empty
  >    Storage Element 2:Full
  >    Storage Element 3:Full
  >    Storage Element 4:Full
  >    Storage Element 5:Full
  >    Storage Element 6:Full


- Determine which directory to restore:
  You need to find out the full Unix directory
  of the data to be restored.
  Example: Let's asume the project directory is
  located at:

      /home/sls/projects/

  and you want to restore the data of customer "Ross".
  Then your "full Unix directory path" is:

      /home/sls/projects/ross/

- Determine where to restore the data:
  You need to restore the data into a place where you
  can access them conveniently. For example, you could
  choose your home directory
```

```
      /home/sls/


- Create a "recovery" directory:
  We recommend you to setup a new directory to place
  the recovered data, so that you don't confuse them
  with the current data:

      /home/sls/recovery.YYYMMMDDD (Year, Month, Day)

  To create this directory type:

  # cd /home/sls/
  # mkdir recovery.YYYMMMDDD
  # cd recovery.YYYMMMDDD


- Now you are prepared for the recovery itself:
  Change to the new recovery directory:
  # cd /home/sls/recovery.YYYMMMDDD

  Rewind the tape to the beginning
  # mt rewind

  Start the recovery process (can take about an hour!)
  Please note that there is NO leading "/" before the
  "home/sls/..." (the full Unix directory path"):
  # tar xf /dev/tape home/sls/projects/ross


- Make the directory readable by the public:
  The restored data are currently only readable for the
  root user. Make them globally readable using:

  # chmod -R go=u /home/sls/recovery.YYYMMMDDD


- Access the restored data:
  Please look for the "recovery.YYYMMMDDD" directory
  in your file server.  You can now access the restored
  data in the same way as your normal data


- Remove the folder after the recovery has been completed:
  Be extremely caucious with this command, it may delete
  your entire server (for example, if you only add a space
  between "home/" and "sls"...)

  # rm -r /home/sls/recovery.YYYMMMDDD
```

Outdated! The contents of this manual are not relevant anymore. Please see the "PO-Unix-Simplified-Install-Guide" for Linux installation instructions.

Project/Open

Avda. Felix Millet 45
08338 Barcelona, Spain
Tel.: +34 609 953 751
Fax.: +34 93 741 1235