The CLUE Administration Handbook

The CLUE Team

March 04, 2003

Contents

1 Contact Information

1.1 CLUE Contact List:

Administrator Name	Dept.	Email	Telephone	Home	Cell	Responsibilities
Matt Bradshaw	ECSS	mbrad1@iastate.edu	4-5313	597-4196	N/A	
John Dickerson	ECSS	jedicker@iastate.edu	4-5376	233-9835	515-450-2942	
Steven Kovarik	CSG	skovarik@iastate.edu	4-1583		515-290-4870	
Steve Kroeger	CSG	skroeger@iastate.edu	2-8139			
Joe Mesterhazy[1]	CSG	jmesterh@iastate.edu	2-8139	268-3232	515-460-1871	
Don Schlagel	ChE	schlage@iastate.edu	4-4919	233-4144		ChE Dept.
Ben Vaughan	ECSS	benvon@iastate.edu	4-1629	233-2837	515-231-9135	
Jim Wellman	AeroE	wellman@iastate.edu	4-0542			

2 About the CLUE network

A new philosophy of computer administration has formed in recent years in the College of Engineering. Just a few years ago, each computing support team in every department was an island in and of itself. Very little communication with other departments resulted in each department duplicating effort and solving the same issues over and over again. During this time a small project spear-headed by ECSS (Engineering Computing Support Services) and ECpE to integrate UNIX logins across the College of Engineering was born. This project, dubbed 'CLUE' (College Level UNIX Environment) soon grew into something much more.

2.1 In the beginning..

Before CLUE existed, every departmental computing support group on campus was isolated. Departments that were large enough to need their own computing support team depended on their own knowledge and experience to guide them on their day-to-day activities. Each department would come up with their own methods of login, printing, authentication, file serving.. and became comfortable with what they were providing. As time progressed, this became a 'comfort zone' that was seldom deviated from. Unfortunately, due to departments going about their business in their own way, computing facilities would vary widely from department to department. For example, faculty in one department may have support for Windows XP, but in another department the administrators would support Windows NT only. Students in one department had access to Linux, while in others they only had access to the Windows platform. Unfortunately this became problematic, because as the computing support personnel tended to stay within their own department, the faculty and student body tended to be very mobile. Unknowing to the administrators, this lead to an unspoken hostility between the support personnel and the very people they were supporting. Why? A faculty or staff member may ask to have Windows XP installed, only to be turned down by his computing support personnel because it was "unsupported" (eg. out of their comfort zone). This would often lead to resentment, because this faculty or staff member would know full well that his peers in other departments would not have been turned down with the very same request.

And so this is how it was for a good many years within the College of Engineering. Each department built their computing support platform in completely different ways, with little to no consistency.

2.2 The meme spreads..

As word about the CLUE network spread, more and more communication between computing support groups began to take place. Mailing lists were replaced with an IRC-style live chat, and suddenly administrators were able to communicate in real time. We were no longer alone! As the number of departments joining CLUE increased, the 'brain trust' grew... and with it so did the comfort zones. Time that was once wasted searching for solutions online was almost always a problem another administrator had already solved.. a simple question to the CLUE chat window resulted in an instant answer from another CLUE administrator. Time once spent tracking down problems could instead be spent on coming up with new ideas or implementing new technologies, and implementations they had once said were unsupported were quickly put into production.

2.3 CLUE today

The CLUE network as it stands today is no longer limited to the UNIX workstations within the College of Engineering. Now all UNIX and Windows clients use the same login and file space. We created an active directory domain called ENGR, every department has the option of adding their own server, and all CLUE administrators have administrative access to this domain. Every administrator, for better or for worse (but almost always for the better), has equal control over the domain. There is no hierarchy, no secret passwords, no information hoarding, and no central authority figure. This method of administration is unique on the ISU campus and is something we are very proud of. The CLUE network provides services no other central computing group on campus can match, including transparent unified file space between UNIX and Windows, strong authentication, encrypted remote file access, remote desktop login from off-campus to both Windows and UNIX, and a feature-rich web hosting environment that will soon support WebDAV and PHP scripting. None of these features are requirements if one wishes to join the CLUE network. The CLUE network is not defined by the computers or software applications it provides, but by the spirit of cooperation and open communication that makes it so rewarding.

We believe that the CLUE network represents the future of computing resource administration, tearing down the walls isolating computing support groups on campus and building new lines of communication and camaraderie for the benefit of all departments on campus.

3 CLUE Overview

The CLUE Network was designed with a few basics goals:

- The framework should be open enough to accomodate a wide range of systems and technologies. The most appropriate technology for the problem at hand should determine what the network supports, not the other way around.
- As much as possible, users should be presented with a consistent experience, regardless of the platform. Moving from one platform to another should feel as seamless as possible.
- The techniques for administering the systems should be the same for similar systems but flexible enough to handle differing needs.

To these aims, the CLUE Network is actually a collection of components that work together to create a unified computing environment. Some components are central to the overall design of the system, some serve a supporting role. We can say that the CLUE components are either *user services* or *adminstrative tools*.

The user services consist of:

- User authentication
- Network file sharing and printing
- e-Mail
- Web servers

The administrative tools are:

- User management (ComputeAdmin)
- Platform build procedures and software management
- Configuration management
- Instant messaging
- Trouble ticket management (RT)

Except for perhaps printing, all of these services and tools are universal across the network.

3.1 User Authentication and File Sharing

The core CLUE user services are the authentication and file sharing environments. CLUE is a hybrid system designed to support Windows and UNIX systems equally in their native file sharing and authentication protocols. For instance, all CLUE users authenticate Kerberos servers running on top of Microsoft Windows 2000 domain controllers. Since Kerberos is the preferred authentication method for both Windows and UNIX, running Kerberos on the Windows domain controller allows both UNIX and Windows platforms to use the same authentication environment. Also, having a MS Active Directory allows Windows clients to take full advantage of built-in file sharing and access control features supported natively by Windows. But since most UNIX systems do not yet work seamlessly with MS Active Directory, NIS is used to provide basic home directory and user account information to UNIX clients.

Most file servers in CLUE are Linux servers that support NFS file sharing to UNIX clients and SMB/CIFS file sharing to Windows clients. File sharing to Windows clients is handled by Samba. This, along with Kerberos, provides a "single sign-on" experience for users allowing them to use the same login password to access the same data regardless of the type of platform they are logged into.

There are two Windows 2000 domain controllers that handle the Active Directory and Kerberos role. The machines (called **w2kdc1** and **w2kdc2**) are housed in 208 Marston Hall and are operated mainly by ECSS. The name of the Windows Active Directory domain is **ENGR.IASTATE.EDU**, though the Windows domain is simply **ENGR**.

A master NIS server provides the usual UNIX *passwd*, *netgroup*, and automounter tables to UNIX clients. NIS is not used for authentication. All authentication is done by the Windows domain controllers. The NIS domain is called *clue*. The master NIS server is **deimos.ee.iastate.edu** and is managed by CSG in 2101 Coover Hall.

There are over a dozen file servers that are part of CLUE. Most are Linux file servers running Samba 2.4 and NFS v3. There are also several Windows file servers that serve Windows clients only. The decision to house some users in Windows-only file servers is done primarily for legacy reasons. Some users' home directories existed before the Linux servers were available and have not been moved to the Linux file servers. Those users also frequently do not use UNIX/Linux so it is not as critical to move them to the Linux server. The file servers are maintained by several groups in the college:

The location of a user's home directory is determined largely by the department they are in. One aspect of the CLUE structure is that individual departments are free to set up their own file servers as the need arises. Some departments choose to operate their own file servers. Some choose to use common file servers available to the whole college. ECpE and ChemE operate their own Linux file servers for storing data for users (faculty and students) in their departments. Other departments frequently use college-wide servers operated by ECSS.

The process for connecting a user with the appropriate file server where their data resides varies depending on whether the platform is Windows or UNIX/Linux. When logging in to a Windows client, the system queries the Windows Active Directory to get the network path to their home directory. On a UNIX/Linux client, the NFS automounter queries the NIS *auto.home* database for the network path to the user's home directory and

mounts it during login. The figure below illustrates the relationship of the various authentication, database, and file servers for users on the network.

3.2 e-Mail Services

In addition to the core of the system are more modular parts of the environment. There are two electronic mail environments: a Microsoft Exchange environment that is used by faculty and administration staff in the collge, and an IMAP server that is used by some students and faculty. The Exchange environment serves email, personal and group calendars, and address books. The e-mail address suffix for the Exchange server is @engr.iastate.edu. The IMAP server serves mail only. Its address suffix is @eng.iastate.edu.

Users who choose to receive mail on either the Exchange server or the IMAP server usually forward their mail from the campus e-Mail system managed by AIT to the appropriate address on either of the two servers.

3.3 Web Services

There are several web servers available to users in the college. The main college web server (http://www.eng.iastate.edu) is an IIS server that serves mainly general information web pages for the College of Engineering. Some departmental web pages are also served by that server.

4 The 'K' Desktop Environment

4.1 Why KDE?

Towards the end of 2001, it was decided that a new common desktop was needed for the various UNIX workstations administered by CLUE. The obvious choices were GNOME and KDE. GNOME had been experimented with prior to this discussion, but in the end KDE was chosen because it was the only desktop known to compile easily across HPUX, Solaris, Linux, and IRIX (the four supported UNIX platforms at the time). KDE on HPUX was never fully realized as the platform was dropped, and KDE compiled on IRIX after a bit of a tussle. In contrast KDE compiles smoothly and easily on Solaris and Linux. KDE provides a powerful, easy-to-use desktop for UNIX workstations in CLUE. Although GNOME is not officially supported by CLUE, it should also available on workstations should users wish to use it. This is easily accomplished since GNOME comes with Red Hat Linux and Solaris in a pre-packaged form.

4.2 Compiling KDE

Like all CLUE applications, KDE should always be built from a source RPM. The target directory is /usr/local/kde. It is installed into its own tree because of its size, and to make the installation location consistant between Solaris and Linux. The source RPMS are located in /remote/apt/redhat8/SRPMS.kde-clue for Linux, and /remote/apt/solaris8/SRPMS.clue for Solaris (this should be made into its own APT tree like Linux.) If you have questions about the KDE packages, Joe is the current maintainer so you should send them his way.

4.3 CLUE customizations to KDE

The KDE that comes with recent builds of Red Hat Linux is a bastard form modified by Red Hat and should not be installed, as it installs into the /usr prefix and breaks any local CLUE installation in /usr/local/kde. The CLUE KDE configuration files are stored in /usr/local/kde/clue, which is maintained in the clue-kde-config RPM.

4.3.1 KDE configuration

Unlike Microsoft Windows and GNOME, KDE does not use a central registry for application settings. Each application in KDE keeps its settings in an rc file. For example, kmail keeps its settings in a file called kmailrc. Settings can be set on a site-wide and a per-user basis. This is accomplished via an environment variable called $KDEDIRS which is set by /etc/profile. Like the $PATH environment variable, $KDEDIRS contains a colon-delimited list of directories. The $KDEDIRS is also implied to begin with $KDEROOT (/usr/local/kde) and end with $KDEHOME (~/.kde31).

When a KDE application starts up, it searches for configuration information in each directory in $KDEDIRS, with specific settings from later configuration files overriding previous ones. CLUE currently uses three

paths, in order: /usr/local/kde, /usr/local/kde/clue, and ~/.kde31. Sine the first and last are hard-coded in $KDEDIRS, KDEDIRS is simply set to /usr/local/kde/clue. /usr/local/kde contains the configuration file defaults set by KDE.org. /usr/local/kde/clue contains application customizations for the CLUE network, and ~/.kde31 is the user-specified application configuration. Keep in mind configuration files override on a **per-setting** basis, not per configuration file. For example, if there is a kmailrc file in /usr/local/kde/clue/share/config/kmailrc, and one in ~/.kde31/share/config/kmailrc, any setting in /usr/local/kde/clue/share/config/kmailrc that is not restated in ~/.kde31/share/config/kmailrc will remain unchanged.

4.3.2 Locking down KDE cofiguration

Helpful url: http://www.linux-mag.com/2002-11/kde_01.html

What if you do not want users to change a certain setting within KDE? For example, you do not want screen saver locking enabled. In the file kdesktoprc you will find this option:

```
[Screensaver]
Lock=true
```

You can put this statement into /usr/local/kde/clue/share/config/kdesktoprc, changing Lock to 'false'. However this just sets the default behavior. Users will still be able to turn it back on. How do we prevent this? KDE provides a configuration locking mechanism called KDE kiosk. This is very similiar to 'policies' in the Windows world. All configuration settings can be marked as 'immutable', that is, once this setting is set, it cannot be changed. This can be accomplished by simply adding a [$i] to the value. For example, to prevent anyone from turning on screen saver locking, add this to the kdesktoprc file in /usr/local/kde/clue/share/config/kdesktoprc

```
[Screensaver]
Lock[$i]=false
```

It is now no longer to turn on screen saver locking, despite what the user has set in ~/.kde31/share/config/kdesktoprc, because /usr/local/kde/clue is in the KDEDIRS path before ~/.kde31.

There are also other neat features of the KDE kiosk mode, such as inserting the values of environment variables in configuration files. This is used in the CLUE configuration of kmail. The setting login is set to login[$e]=$USER. This sets the login variable to the username of the current user. Using this technique, along with setting other defaults, kmail is ready for use on the first launch, no user intervention is required.

4.3.3 The CLUE KDE configuration RPM

Files in /usr/local/kde/clue are installed via an RPM called "clue-kde-config". There is no SRPM, as the tree is the source. In /usr/local/kde/clue you will find a SPEC file called 'clue-kde-config.spec'. To modify this tree, simply make the changes you like, increment the version number in clue-kde-config.spec, execute "rpm-build -bb clue-kde-config.spec" and copy the new RPM to the APT tree in /remote/apt/redhat8/8/RPMS.kde-clue. The SPEC file is included in the RPM for the next round of modification.

4.4 KDE 'desktop' files

Applications are associated with an icon and a mimetype (or file extension) through the use of 'desktop' files. These are the same format as used by GNOME and FreeDesktop.org. Here is an example of a desktop file:

```
[Desktop Entry]
Exec=/usr/local/bin/acroread
InitialPreference=7
MimeType=application/pdf
Name=Acrobat Reader
GenericName=PDF Viewer
Terminal=false
Type=Application
Icon=acroread
```

The above example creates a desktop file for Acrobat Reader. The `InitialPreference` token sets Acrobat Reader to be the default viewer for application/pdf. Because many different applications can open up the same type of file, the `InitialPreference` token specifies this application is of priority seven. If no other desktop files for this mimetype have an `InitialPreference` higher than seven, then Acrobat Reader will become the default. Seven is usually a safe bet for making a desktop file the preferred application for a certain mimetype.

5 FEP – Frequently Encountered Problems

5.1 ISSUE: You need to add a new Linux machine to the CLUE environment.

Solution:

- Obtain fully qualified hostname from DNS
- Ssh to deimos.ee.iastate.edu
- Edit /etc/nis/netgroup and add the host into one of the smaller blocks in the file. **Follow syntax precisely.**
- Do a ypmake
- Use the kickstart boot floppy (or CD) to boot your system and build it.

5.2 ISSUE: Machine will not allow local login by anyone except root.

Solution: check time sync

- On Winders, reset clock to within 3 minutes of time.iastate.edu
- Linux: *rdate -s time.iastate.edu*
- SUN*: ntpdate*

5.3 ISSUE: Cfengine build fails on system rebuild.

Solution: You most likely forgot to remove the machine's ppkeys from deimos /var/cfengine

5.4 ISSUE: You need to determine if an RPM[2] is installed on your machine.

Solution: RPM -q packagename

5.5 ISSUE: You need to create an NIS mapped share on your linux server.

Solution:

- First create the dir (e.g. /home/scratchdir)
- Edit /etc/exports and add the share info

[2] Or what version of the RPM is installed.

```
e.g.
/home/scratchdir, @client(rw)
/home/scratchdir, @admin(rw)
/home/scratchdir, @server(rw)
```

- Do an exportfs -r
- On Deimos edit the /etc/nis/auto.remote file and add the new mapped share If needed.
- Add an NIS group on Deimos (edit the /etc/nis/group file and add the appropriate group members for accessing the NIS share)
- Do a ypmake
- Share should be working in /remote/sharename

5.6 ISSUE: You need to create a samba share for windows users to access a directory on your Linux server.

Solution:

- create the directory which will be shared.
- edit the /etc/samba/smb.conf file to add a share section similar to the one below.

```
[sharename]
comment = this share is an example
browseable = yes
writable = yes
path=complete path to share directory
```

- Note that other options can be set (such as restricting write access to a user or group of users). This should be covered in the chapter covering SMB

5.7 ISSUE: Users cannot login to windows or UNIX machines, but other users can. [3]

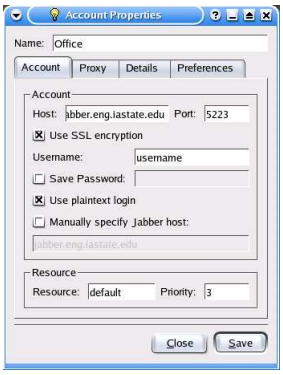
Solution:

- Add quota using computeadmin pages so user can login and reduce usage.
- Fix profile by moving .ntprofile to .ntprofile.old in their homedir on the samba server.

[3] A corrupt profile will exclude windows logins. Running over quota will exclude both windows and UNIX logins.

5.8 ISSUE: Psi will not connect to jabber.eng.iastate.edu

Solution: Be sure to enable the "use SSL encryption" and "use plaintext login" checkboxes in the account setup dialog. (see figure below)



5.9 ISSUE: You need to run an X-based application from one UNIX machine, while logged into another UNIX machine (or Windows machine with an Xserver installed).

Solution:

- on your machine do: *xhost + machinename.subnet.iastate.edu*, where *machinename* is the name of the machine that has the desired application.
- ssh to machinename.subnet.iastate.edu
- do: *export DISPLAY=yourmachine.yoursubnet.iastate.edu:0.0*
- run the X-app

5.10 ISSUE: Idle connections in terminal services cache old password preventing U: drive access. Error message: NT_STATUS_WRONG_PASSWORD

Solution:

- clear the user's idle sessions from TS Server and try again.

5.11 ISSUE: Error message on Windows machine stating the ENGR domain is not available.

Solutions:

- Machine may not be in the DHCP server's list of IP clients
- WINS and DNS settings may be faulty, or servers are inoperable.
- Bad network connection on client machine.

5.12 ISSUE: Windows print queue stuck on a job that cannot be deleted.

Soultion:

- net stop spooler
- net start spooler
- check out print accounting software to ensure it still functions

5.13 ISSUE: You need to determine the linux kernel version you're running on a specific machine.

Solution:

- uname -a

5.14 ISSUE: You need to stop the X server to re-configure things.

Solution:

- Use Ctrl+Alt+F2 to get a plain terminal.
- Login as root and issue a telinit 3
- run redhat-config-xfree86
- after the configuration is done run telinit 5 to restart the X Server
- Ctrl+Alt+F7 will take you back to the X server terminal

5.15 ISSUE: Linux server is being unresponsive to the network. Load is ok, net settings are ok, but nothing can connect.

Solution:

- Check to make sure scroll-lock isn't on in a VT that shows sylog messages. The buffer fills up and kills the box.

5.16 ISSUE: You need to move a user's home directory from one server to another.

Solution:

- Move files to the new server location.
- Edit the auto.home file on Deimos.
- Edit the AD pointers for the user on the ComputeAdmin pages.

6 Connecting linux computers via serial line.

Basic method for connecting linux computers via a serial connection.

6.1 Host Computer.

6.1.1 Editing the /etc/inittab file.

The Identifier should reflect its use. A typical identifier for the first serial port would be 's1'.

Runlevels indicates which runlevel you want the port to be active. So if you want the serial port active on runlevels 3 and 5 the entry would look like this "s1:35:..."

Action tells init how to handle the executable. There are several options for this field, but the only one we'll need is 'respawn'. This tells init to restart the executable if it dies for any reason. So now our command looks like this "s1:35:respawn:..."

Executable is the program or process you want to run on the serial port. For linux use /sbin/agetty. Now we have "s1:35:respawn:/sbin/agetty..."

Baud-Rate tells the device how fast to communicate over the serial line. A safe speed is 9600. You can go higher if you wish, but you'll need to research available speeds and test the quality of the connection. So we're up to "s1:35:respawn:/sbin/agetty 9600..."

Device tells init which device to make all this active on. For linux the serial ports correspond the devices /dev/ttyS* beginning with ttyS0 for the first serial port. Our final command is "s1:35:respawn:/sbin/agetty 9600 ttyS0" This should get a serial console up and running on the computers first serial port at run-levels 3 and 5. Most setups include run-levels 2 and 4 as well so "s1:2345:respawn:/sbin/agetty 9600 ttyS0" is common.

For the serial connection to become active you'll need to run "telinit q" which tells init to reexamine the inittab file or switch to a run-levels. Rebooting will work too.

6.2 Client Machine.

6.2.1 Configuring the client machine.

On the client machine you'll need to run a terminal emulator to be able to connect the host. Minicom is included with linux, but isn't the most user friendly to setup. Microcom is available via download at http://microcom.port5.com and is more user friendly for day to day use.

Inorder to connect microcom to the serial port you run microcom -D[path-to-device] Example "microcom -D/dev/ttyS0" to connect microcom to the first terminal port (notice there isn't a space between -D and the device).

7 How to Recoveer Data from Floppy Disks in Linux

User has a floppy disks and recently discovered that the disk was corrupted and he could no longer be mounted under Linux. How cann the disk content be accessed using the command dd on linux?

With some difficulty.

In general I would not recommend mounting floppy disks as file systems under Linux, even though there is software that allows you to do it. This is mainly because the Unix file systems were written with more reliable media in mind than floppy disks can ever be, and also because the floppy disk drives and media are now so cheap that production tolerances appear to have been relaxed. Where it was a rare occurrence to have floppy disk errors 10 years ago, they are becoming quite common.

Thus it is better to use mtools or dd to write to floppy disks, and to use a loopback file system to manipulate a disk image with normal file system utilities.

Once a floppy disk is no longer readable, the following should be done:

- Try another floppy disk drive, this is often the quickest way.
- Use dd to retrieve bits of data:

  ```
  dd if=/dev/fd0 of=file1 bs=18k
  ```

  18 kilobytes is the "cylinder size" on the common "high density" 1440 kilobyte floppy disk, i.e. 2 tracks of 18 sectors of 512 bytes each. If you have "extended density" (2880 kilobyte) or "double density" (720 lilobyte) floppy disks, adjust the numbers accoordingly.

  The command will abort with something like:

  ```
  15+1 records in
  15+1 records out
  ```

  Now re-issue the dd request, we need to find out how much data we got from track 15 (luckily it stats counting at 0, so 15 tracks read means track 15 is bad. We now use 512 byte blocks. Calculate 15 * 18 * 2 = 540

  ```
  dd if=/dev/fd0 of=file2 bs=512 skip=540
  4+0 records
  in 4+0 records out
  ```

  so sector 545 is bad ...

  ```
  dd if=/dev/fd0 of=file3 bs=512 skip=545
  0+0 records in
  0+0 records out
  ```

  as is 546 ...

- Now build a single disk image, substituting zeros where you found bad sectors:

```
rm image
touch image
dd if=file1 bs=18k count=15 >> image
dd if=file2 bs=512 count=4  >> image
dd if=/dev/zero bs=512 count=2 >> image
...
```

  This is necessary to avoid duplicating the tail ends of abortive reads and the data from the next read.

- You now should have a disk image of 1474560 bytes. Now mount it. If the file system was cheeckable (like e2fs), then check it first:

```
fsck -t fstype image
mkdir fs
mount -t ext2 -o ro,loop image fs
```

- Now extract the files from fs, and check if there are blocks of zeros embedded in them. If you find blocks of zeros, recreate the data by other means.

- You can also write this image to a floppy disk:

```
dd if=image of=/dev/fd0
```

In many ways, the MSDOS file system is simple, and more daata will be retrievable as long as the FAT has survived. This will be the case if the first track is copied successfully.

8 EE Servers and Clients

8.1 Servers

master.ee.iastate.edu -

Master is a linux server that is the center point for cfengine and apt.

logic.ee.iastate.edu -

Logic, linux server. All syslogs are sent to logic. Logic also houses home directories for VLSI grad students.

dcnl.ee.iastate.edu -

Dcnl houses home directories for DCNL grad students.

deimos.ee.iastate.edu -

Deimos is the primary NIS server for EE. It also houses some of the home directories for undergraduate students.

phobos.ee.iastate.edu -

Phobos is a secondary NIS server. It also houses some of the home directories for undergraduate students.

io.ee.iastate.edu -

Io is a secondary NIS server. It also houses some of the home directories for undergraduate students.

tux.ee.iastate.edu -

Tux is the ssh gateway to most servers and clients in EE. Most unix clients will only accept ssh conections from tux.

laplace.ee.iastate.edu -

Laplace is an antiquated NIS server. Laplace is the NIS server for power5 and tesla. Two HP machines.

faith.ee.iastate.edu -

Faith servers remote apps over nfs (cadence, synopsys, etc.).

europa.ee.iastate.edu -

Europa is the faculty home directory server.

linux-1.ee.iastate.edu -

Linux-1 is our first remote access server.

linux-2.ee.iastate.edu -

Linux-2 is our secondary remote access server.

pluto.ee.iastate.edu -

Pluto serves http://clue.ee.iastate.edu

sunserv1.ee.iastate.edu -

Sun licence server.

jupiter.ee.iastate.edu -

Jupiter is a remote access sun X server.

saturn.ee.iastate.edu -

Saturn is a remote access sun ssh server.

charity.ee.iastate.edu -

Charity is our remote access MS Windows server.

hope.ee.iastate.edu -

Hope is our Active Directory and print server. Most print jobs go through hope.

designhost.ee.iastate.edu -

Designhost servers virtuals linux machines for serior design projects. design-1 through design-10.

8.2 Coover Unix Clients

8.2.1 Coover 1331

battery.ee.iastate.edu	capacitor.ee.iastate.edu	crystal.ee.iastate.edu
diode.ee.iastate.edu	inductor.ee.iastate.edu	relay.ee.iastate.edu
resistor.ee.iastate.edu	thyristor.ee.iastate.edu	transformer.ee.iastate.edu
tube.ee.iastate.edu	variac.ee.iastate.edu	jumper.ee.iastate.edu
solder.ee.iastate.edu	ic.ee.iastate.edu	fuse.ee.iastate.edu
led.ee.iastate.edu	heatsink.ee.iastate.edu	socket.ee.iastate.edu
triode.ee.iastate.edu	triode.ee.iastate.edu	zener.ee.iastate.edu
schotty.ee.iastate.edu	piezo.ee.iastate.edu	chip.ee.iastate.edu
solenoid.ee.iastate.edu		

8.2.2 Coover 1316

mauchley.ee.iastate.edu	tukey.ee.iastate.edu	ritchie.ee.iastate.edu
backus.ee.iastate.edu	coulomb.ee.iastate.edu	eckert.ee.iastate.edu
forrester.ee.iastate.edu	hopper.ee.iastate.edu	thompson.ee.iastate.edu
turing.ee.iastate.edu	zworykin.ee.iastate.edu	joy.ee.iastate.edu
kernighan.ee.iastate.edu	olsen.ee.iastate.edu	packard.ee.iastate.edu
robinson.ee.iastate.edu		

8.2.3 Coover Carver Lab

carver1.ee.iastate.edu	carver2.ee.iastate.edu	carver3.ee.iastate.edu
carver4.ee.iastate.edu	carver5.ee.iastate.edu	carver6.ee.iastate.edu
xga.ee.iastate.edu	mga.ee.iastate.edu	

8.2.4 Coover Power Lab Solaris (Jack London Lab)

hoover.ee.iastate.edu	chernobyl.ee.iastate.edu	indio.ee.iastate.edu
colmubia.ee.iastate.edu	louisa.ee.iastate.edu	onofre.ee.iastate.edu
altamont.ee.iastate.edu	diablo.ee.iastate.edu	power5.ee.iastate.edu

8.2.5 Coover 3223 Solaris Lab

univac.ee.iastate.edu	vax.ee.iastate.edu	eniac.ee.iastate.edu
altair.ee.iastate.edu	abc.ee.iastate.edu	pdp.ee.iastate.edu
cbio1.ee.iastate.edu	cbio2.ee.iastate.edu	cbio3.ee.iastate.edu
cbio4.ee.iastate.edu	cbio5.ee.iastate.edu	cbio6.ee.iastate.edu
cbio7.ee.iastate.edu	cbio8.ee.iastate.edu	cbio9.ee.iastate.edu
cbio10.ee.iastate.edu	cbio11.ee.iastate.edu	

8.3 Durham Unix Clients

8.3.1 DCNL Solaris Clients

hippi.ee.iastate.edu	kona.ee.iastate.edu	fddi.ee.iastate.edu
arcnet.ee.iastate.edu	java.ee.iastate.edu	latte.ee.iastate.edu

8.3.2 Comserver Solaris Clients

noyce.ee.iastate.edu	pease.ee.iastate.edu	ginger.ee.iastate.edu
cinnamon.ee.iastate.edu	cga.ee.iastate.edu	eesun1.ee.iastate.du

8.3.3 Durham 310 Linux Clients

mauve.ee.iastate.rdu	gray.ee.iastate.edu	sage.ee.iastate.edu
ginseng.ee.iastate.edu	oregano.ee.iastate.edu	basil.ee.iastate.edu

8.3.4 Durham 324 Linux Clients

karnak.ee.iastate.edu	cairo.ee.iastate.edu

Index