

# **Advanced Topics in Systems Administration Projects Collection**

**For CSCI5360, Spring 2003**

## Table of Contents

<b>Introduction</b> .....	<b>3</b>
<b>Operating Systems</b> .....	<b>4</b>
Bastille Linux Installation (Hankerson).....	5
Building a Production Kickstart Server for Red Hat Linux 8.0 (Hankerson, Nielson) .....	10
CODA File System Installation (Divvala).....	17
Configuring X-Windows for Open BSD (Berry).....	25
FreeBSD Installation (Hickman) .....	43
OpenBSD Installation (Berry) .....	45
RIS Kickstart.....	47
Automating System Setup Using RIS, Kickstart, and PXE (Nielson).....	47
RIS Kickstart for CSCI 3400 (Hankerson, Nielson).....	51
RIS Kickstart for CSCI 4417 (Hickman, Berry, Buck) .....	56
Setting up Red Hat Linux User Environments (Fritts) .....	62
Solaris Installation (Franklin) .....	73
<b>Security</b> .....	<b>79</b>
AIDE Installation (Wambaugh).....	80
Deep Freeze Installation (Hankerson) .....	84
Firewall Scheduling Tool (Baradi, Chua, Hankerson).....	90
Honeypot Bakeoff (Fritts).....	93
Open AFS Kerberos Integration (Chua, Divvala).....	104
OpenBSD Firewall Installation (Baradi, Chua, Hankerson).....	108
Security-Enhanced Linux Installation (Buck) .....	111
Tripwire Installations .....	116
Tripwire for Linux 8.0 Installation (Frazier) .....	116
Tripwire Open Source, Linux Edition Installation (Fritts) .....	124
<b>Applications</b> .....	<b>129</b>
Big Brother Installation (Hickman) .....	130
BrightStor ARCserve Installation (Nielson).....	138
Bugtracker Bakeoff (Amanda Hickman).....	142
Bugzilla Installation (Hickman).....	150
CVS Installation (Nyabando).....	152
LDAP Installation (Divvala).....	155
MySQL, Apache, PHP Installation for Linux (Berry).....	158
OSCAR Installation (Franklin, Gudepu, Nyabando, Wambaugh) .....	172
Shavlik HFNetChkLT Installation (Nyabando).....	177
Sun One Active Server Pages Installation (Berry) .....	182
Virtual Network Computing for Linux (Buck).....	190
<b>Utilities</b> .....	<b>194</b>
Building an RPM for PLT (Wambaugh) .....	195
Command Builder (Buck).....	198
CVSFE (Wambaugh) .....	220
etsumail2passwd.bash (Nielson).....	240
Ghost AutoInstall for Configuration Management (Buck).....	243

K12 Linux Terminal Server Installation (Chua).....	247
Lab Application Scheduling Utility (Berry).....	249
Network Monitor Bakeoff (Todd Franklin).....	250
Web Application Stress (WAS) Tool (Nielson).....	262
Webmin (Frazier).....	266
zenTrack Installation (Nyabando).....	271
<b>E-Mail.....</b>	<b>274</b>
IMail Installation (Buck).....	275
Liberum (Nielson).....	278
Microsoft Exchange (Berry, Chua, Divvala, Nielson, Nyabando).....	281
Postfix, IMAP, Pine Mail Server (Franklin, Fritts, Hankerson, Hickman).....	284
Sendmail, Cyrus Mail Server (Buck, Duan, Frazier, Simons, Wambaugh).....	287
<b>Case Studies and Other Projects.....</b>	<b>290</b>
Building Cables for Wilson Wallace (Nyabando).....	291
LDAP Namespace Design (Fritts, Simons).....	294
System Administration Practices for Bush Hog, LLC (Nyabando).....	303
<b>Lisa Write-up.....</b>	<b>307</b>

## Introduction

This book is a compilation of the projects done in CSCI 5360 (Advanced Topics in System and Network Administration) in the Spring of 2003 by graduate students at East Tennessee State University. The contributors have a range of backgrounds: professional Systems Administrators (SAs) as well as graduate students with no hands-on systems administration experience have taken the course to deepen their knowledge and understanding of the issues involved in Systems Administration. The prerequisite for this course is a one-semester course (CSCI 4417/5417) in the fundamentals of System and Network Administration, where students learn the basics of systems administration, including operating system installation and configuration, user and resource management, as well as the operation of many common components of a networked infrastructure.

As part of this advanced course (CSCI 5360), each student was required to complete a number of projects of their own choosing. The interests and backgrounds of the students are well-reflected in the choice of projects. Some students chose to install and examine operating system distributions they were not familiar with, or to write code for systems administration tools. Still others chose to look at applications or systems that are helpful to SAs. Other topics were covered as well, showing the breadth of systems administration.

The students overwhelmingly viewed the projects as a positive component of the course, and many have requested that the entire set of project write-ups be made available. This publication is the result of their efforts, and my hope is that you enjoy and learn from the papers.

I'd particularly like to thank Steve Fritts for serving as the editor for this publication. Working as an editor is not something normally encountered in a Computer Science graduate program, and Steve has done a fantastic job ensuring the papers were submitted properly and in working with students (and me!) to get a great document assembled. Many thanks, Steve.

Steven Jenkins  
June 12, 2003

# *Operating Systems*

**Product:** Bastille Linux

**Product Type:** Red Hat Linux Hardening Tool

**Author:** Mario Hankerson

---

### Problem Background

The evaluation of Bastille Linux as a security add-on was chosen to determine security vulnerabilities and to measure the effectiveness of locking down a Red Hat Linux 8 distribution. As a novice Linux administrator, understanding what constitutes a threat and why becomes alarmingly obvious after a system has been compromised. The Bastille Linux tool seems to be essential to limiting and understanding system compromises for system administrators of all skill level. Overall, Bastille Linux is a security WYSIWIG for Red Hat 8 security challenged users to secure their systems.

The main goal of using Bastille Linux is to ensure that the system is locked down and runs as smoothly as possible, limiting hackers or malicious user's abilities to ruin the user's machines. In turn, this will allow for increased up-time, preventing huge losses for the company or individual system owners in loss of work time.

### Product Placement

Bastille Linux is a set of Perl scripts intended to harden Linux platforms security. Additionally, it is a security tool used to provide increased security and system awareness for Red Hat users. Simply stated, Bastille Linux offers the user a series of security questions to answer.



Figure 1.1: Opening Interface

It then configures their system based on those responses. Bastille Linux literally sets up security parameters for the user so that the system is hardened and not vulnerable. Bastille was designed to educate the administrator of all the security issues involved so as to better protect boxes and administrators. In addition, users can manually harden systems and change configurations--enable or disable system services. Bastille Linux scripts ask a variety of detailed questions; wherein, the developers teach the users. They explain the context of the questions and provide the results of the available answers. Thus, the tool becomes an interactive means of assuring higher levels of security.

The Bastille Linux interactive menu is available in GUI form and the traditional command line view, which increases the tool's attractiveness because of ease of use regardless of Linux skill level. For GUI form `bastille` must be typed at the command line, and for command line view `bastille -c` must be entered.

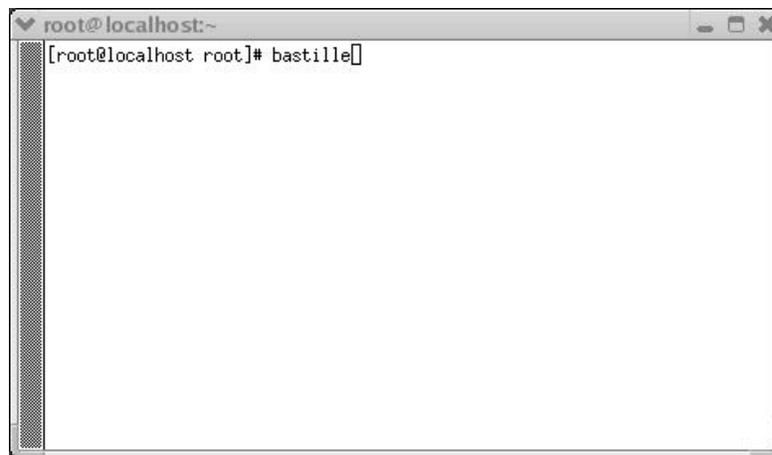


Figure 1.2: Command Line

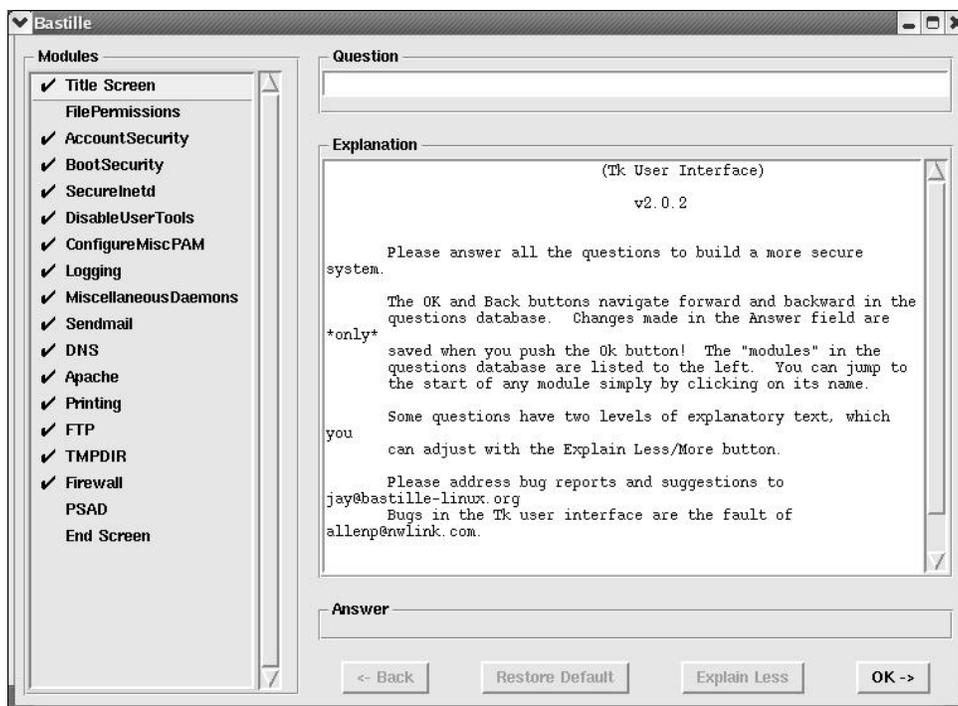


Figure 1.3: GUI Interface

A system administrator or even a home user will inevitably see ways they can tighten down systems for added security and increased optimization that may have been overlooked by using the tool. The use of Bastille Linux demystifies security because it serves as a tutorial while the user is interacting with the tool during system configuration.

### Installation Overview

Bastille Linux is an open source utility and useful documentation is extremely easy to find. Minimum requirements for installation are as follows: a system running Red Hat 8, Perl 5, “enough” ram, and “some” free disk space. Some of the aforementioned requirements could not be quantified due to the lack of documentation specifically stating Bastille’s requirements. In order to install Bastille Linux, the user must be logged in as root since the scripts will change configuration files.

After downloading the Bastille rpm, the user must type `rpm -ivh Bastille-2.0.4-1.0.i386.rpm` at a console or use the GUI rpm management tool in order to install Bastille Linux. If Perl is not available on the machine, the user must download the Perl rpms. In addition, the user must download two Perl modules, which are `perl-Tk-a.b-c.i386.rpm` and `perl-Curses-d.e-f.i386.rpm`; moreover, the user must type `rpm -ivh` along with the rpm file at a console or use the GUI rpm tool that is provided by default with Red Hat. The rpm’s can be found at <http://www.rpmfind.net>.

To begin the program, the user must launch a terminal if they are in X Windows. However, if they are not in X Windows, then they skip this step.

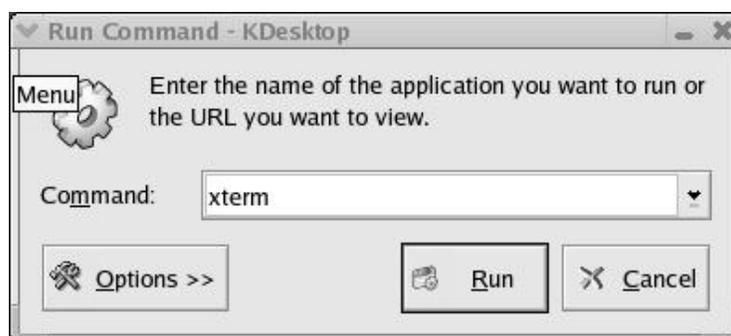


Figure 1.4: Use xterm to launch Bastille

In order to run Bastille the user must type: `bastille -c` or `bastille` at the console to start the hardening scripts. The tool will ask the user several questions pertaining to eighteen modules, such as account, file permissions, ipchains, logs, system boot, and xinetd security.

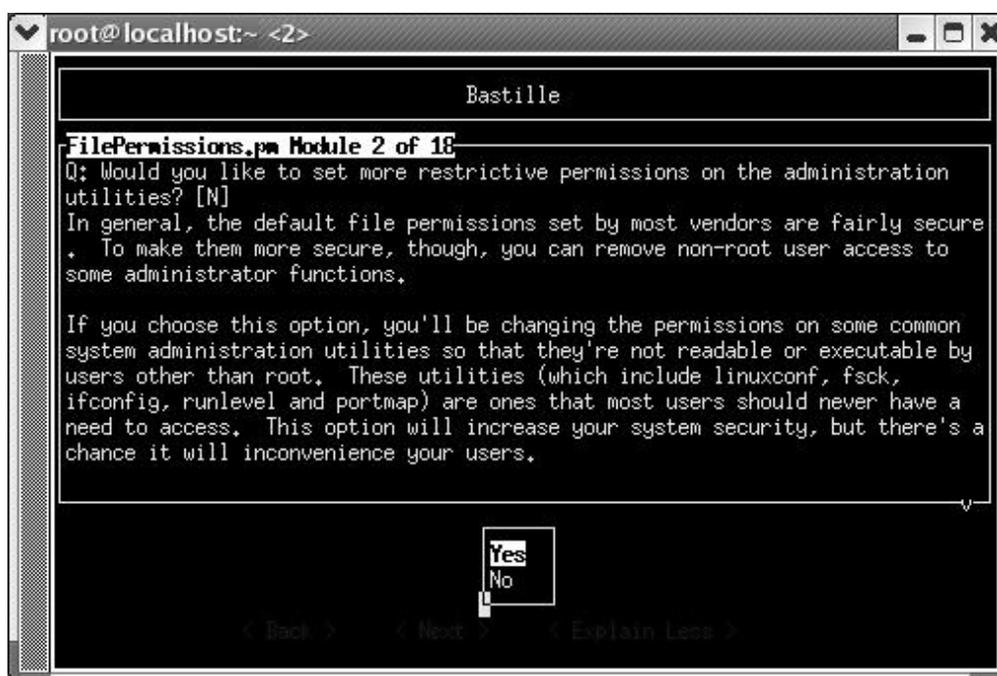


Figure 1.5: Changing file permissions with Bastille

In addition, the interactive nature of the tool will educate the user in regards to what configurations to enable and disable, but ultimately the user makes the configuration decisions. After the user has finished going through the interactive menu, the system must be rebooted for system wide configurations changes to take effect.

### Lessons Learned/Problems

There were no problems with installing or configuring this software. If you do not know much about Linux system security, Bastille Linux will bring you up to speed quickly because it is an educational hardening tool and tutorial. On a cautionary note, if you think your system is absolutely secure, think again because Bastille Linux will prove you wrong. As a first test, two “vanilla” computers were set up with every package installed on them, except the firewall, and

connected via a home network, one with Bastille Linux and the other without. Then both computers were probed using Nessus<sup>1</sup>, a security tool, and the computer without Bastille Linux was shown to have far weaker security. Nessus was able to exploit several known security problems on it. Services that were running by default were the greatest reasons of system insecurity. As a result, system and service configuration files were accessible, thus, from a security standpoint it had no protection whatsoever. The computer that did have Bastille Linux installed was unable to be penetrated and no vulnerabilities were found.

As a second test, my personal Linux system was used as a “guinea pig” already pre-configured to determine how intelligent I was as a security conscience end-user. The only service I stopped from running was the firewall. Amazingly, the system was wide open for all sorts of attacks contrary to my original beliefs, which Nessus later demonstrated. There were services running that had never been used which increased my systems vulnerabilities. This was a slight oversight that can be blamed on the fact that “I didn’t know”, which left my system extremely vulnerable. Rather quickly, I installed Bastille and re-probed my system with Nessus and found no vulnerabilities being reported. The corollary: No one is too good to forget to do something!

### **Final Results/Recommendations**

Bastille Linux works so well that it should be a requirement for all Linux systems before going onto the internet. It is user friendly, easily navigable, and offers sound security that challenges even the most experienced hacker. Overall, the installation was simple and intuitive. As a follow-up to this document, other security tools should be used to determine how effective Bastille Linux is as a hardening tool.

This tool should be a pre-requisite for all personal and corporate systems running a distribution of Linux before the systems are placed on a network. This is due to the fact that many individuals and companies place themselves at risk by not taking proper precautionary measures when it comes to security. With a program like Bastille Linux in place on networks, system disruption can be avoided all together. In addition, the system will teach users more about security and why it is necessary to implement.

### **References**

- Bastille Linux. 11 June, 2003 <<http://www.bastille-linux.org>>  
Lasser, John. "Linux Security: It's Not Just About Security". 8 January, 2000. 11 June 2003 <<http://freshmeat.net/articles/view/141>>  
Raynal, Frédéric. Bastille-Linux. 12 June, 2000. 11 June, 2003 <<http://www.linuxfocus.org/English/September2000/article166.shtml>>

---

<sup>1</sup> Thanks, Rick Simons, for writing a Nessus document.

**Product:** Red Hat Linux 8.0 Kickstart Server

**Product Type:** Server Software

**Authors:** Robert A. Nielson and Mario B. Hankerson

---

## Problem Background

There are many times in the work of a systems administrator where it is necessary to remotely install operating systems and software for multiple computers. Under Linux, this process is referred to as “kickstart”. Red Hat provides all of the basic components necessary to configure a kickstart server on their v.8 CD set. Of course, having the components available does not equate with being ready or able to go ahead and implement a process.

## Project Goals

Hopefully, the following discussion will be reasonably direct and complete in describing the necessary steps to build a production kickstart server from the ground up. This build assumes that there are no servers already present in your environment that could be used for DHCP or other services.

## Walkthrough

The steps necessary to build a production kickstart server include:

- The installation of Red Hat 8
- Configuration of DHCP
- Configuration of DNS
- Configuration of NFS
- Configuration of a TFTP server (this final step is only necessary for designing a system that supports the use of PXE, as well as the ability to start the install via a boot disk)

For purposes of creating the examples in this How-To, a Dell Optiplex GX240 was used as the server. As a follow-up, a Compaq Presario was also tested as the server.

### 1. Prerequisites

The only real hardware requirement, beyond the standards for installing Red Hat 8, is enough free disk space to hold the contents of the Red Hat 8 CDs on one of the partitions. For the client systems, it is required that the systems have a PXE capable network card as well as the basic system requirements for a Red Hat 8 install.

### 2. Server Configuration

On the server’s hard drive, you need to create partitions during the Red Hat installation. Listed below is a sample of what you might create:

```
hda----
|
hda1 /boot ext3 102 MB
hda2 /usr ext3 5718 MB
hda3 /home ext3 2094 MB
hda4-----
|
hda5 / ext3 510 MB
hda6 /var ext3 1020 MB
```

```
hda7 swap 314 MB
```

The server will need a name and IP address. The examples in this How-To use "ks" for the system name and "192.168.100.1" for the IP address. You will need to update these depending on your network and naming conventions. Since, in a single server setting, this box will eventually be running DNS, the DNS server address can be assigned as "192.168.100.1", also. Because this test system was running in a closed environment, the firewall settings were disabled. In a production setting you may want to consider activating the firewall with appropriate settings.

During the installation, there are several options that can be added to the system that may be useful, including X Window; KDE, Editors; Graphical Internet; Text Internet; Server Configuration Tools ; DNS Server; Admin Tools; and Network Servers. Each of the options can be left in its default form except for Network Servers, where the DHCP Server needs to be added to the list. Once the install is complete, you should go ahead and determine the MAC address for your system for later in the setup. You can find this value by going to a \$ prompt and typing:

```
/sbin/ifconfig
```

### 3. Setting up a boot floppy for testing

With the basic box up and running, you'll need to prepare a Linux boot floppy for preliminary testing. To create the boot floppy from a \$ prompt:

```
cd /mnt/cdrom/images
dd if=bootnet.img of=/dev/fd0 bs=1440k
```

### 4. The KickStart config file

The kickstart configuration file, `ks.cfg`, can be created next. Red Hat 8 offers a graphical tool, appropriately named Kickstart, for creating kickstart configuration files. This tool can be accessed by clicking the RedHat icon, then selecting "System Tools". Ideally, the final installation process should require as little user intervention as possible, so each option that the client systems should have after installation should be included. A sample `ks.cfg` file, including hard drive partition information for the client is listed below:

```
#Generated by KickstartConfigurator
#System language
lang en_US
#Language modules to install
langsupport en_US
#System keyboard
keyboard us
#System mouse
mouse genericps/2 --device psaux --emulthree
#System timezone
timezone --utc America/New_York
#Root password
rootpw --iscrypted $1$QIRLoUcn$J.u5ir8QlrqpdTF3tMY1m0
#Install Red Hat Linux instead of upgrade
install
#Use NFS installation media
```

```

nfs --server ks --dir /usr/RedHat
#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes
#Clear all partitions from the disk
clearpart --all --initlabel
#Disk partitioning information
part /boot --fstype ext3 --size=100 --ondisk=hda
part /usr --fstype ext3 --size=1400 --grow --ondisk=hda
part /home --fstype ext3 --size=512 --grow --ondisk=hda
part / --fstype ext3 --size=512 --ondisk=hda
part /var --fstype ext3 --size=384 --grow --maxsize=1024 --
ondisk=hda
part swap --size=256 --grow --maxsize=512 --ondisk=hda

#Use DHCP networking
network --bootproto dhcp
#System authorization information
auth --useshadow --enablemd5
#Firewall configuration
firewall --disabled
#XWindows configuration information
#Probe for video card
#Probe for monitor
xconfig --resolution 1024x768 --depth 16 --defaultdesktop=GNOME
--startxonboot
%packages --resolvedeps
@X Window System
@KDE Desktop Environment
@Editors
@Graphical Internet
@Text-based Internet
@Server Configuration Tools
@Windows File Server
@DNS Name Server
@Network Servers
@Administration Tools
@System Tools

```

## 5. Installation and Workarounds

When PXE is used, a DHCP server is needed to process requests from the client systems. The file `dhcp.conf.sample` can be edited for this purpose. This is simply a matter of changing the IP addresses listed to those appropriate for your network, changing the domain names listed to "kicklab.net", and editing their "ns" host listing to match the server. The line "next-server" was moved below this host entry, as opposed to being a part of the host entry, and a line was added indicating the name of the file that would be passed to the PXE clients when they start the installation process. If you will be using a boot disk instead of PXE, then these final changes are not necessary. The complete contents of the new `dhcpd.conf` file are listed below. If you want to use this file, remember to change the "hardware ethernet" value to the MAC address you found earlier.

```
ddns-update-style interim;
```

```

ignore client-updates;
subnet 192.168.100.0 netmask 255.255.255.0 {
# --- default gateway
option routers 192.168.100.1;
option subnet-mask 255.255.255.0;

option nis-domain "kicklab.net";
option domain-name "kicklab.net";
option domain-name-servers 192.168.100.1;
option time-offset -18000; # Eastern Time
range dynamic-bootp 192.168.100.128 192.168.100.254;
default-lease-time 21600;
max-lease-time 43200;
# we want the nameserver to appear at a fixed address
host kick {
hardware ethernet 00:06:5B:62:81:32;
fixed-address 192.168.100.1;
}
next-server ks.kicklab.net;
filename "pxelinux.0";
}

```

Once the file was edited appropriately, the DHCP service can be restarted using the Services tool or typing `service dhcpd restart` inside of a terminal window.

### 5.1. DNS Server

A DNS server is required for the purposes of this example. This can be configured using the Domain Name Service tool provided by Red Hat. For this example, both forward and reverse lookup zones were added for the domain `kicklab.net`. The forward lookup zone file that was used, `kicklab.net.zone`, is listed here:

```

$TTL 86400
@ IN SOA @ root.localhost (
3 ; serial
28800 ; refresh
7200 ; retry
604800 ; expire
86400 ; ttl
)
kick IN NS 192.168.100.1
@ IN NS 192.168.100.1
kick IN A 192.168.100.1
ks IN A 192.168.100.1

```

### 5.2. NFS Server

The next step deals with configuring the NFS server. Since NFS was included in the Red Hat installation, all you should need to do is edit the `exports` file and starting the service. Of course, before making the folder accessible, it makes sense to actually have a location set aside specifically for kickstart to use. A directory was created under `/usr` that was simply called `RedHat` by typing `mkdir RedHat` while in the `/usr` directory. The contents of the three Red Hat 8 CDs were then copied directly into this location. The copying can be done from the GUI or from the command prompt by typing `cp /mnt/cdrom/* /usr/RedHat` for each CD. Once the files were in place, the NFS tool supplied by Red Hat was used to set this directory for use by NFS. The resulting `exports` file was the single line shown below:

```

/usr/RedHat *(ro,sync)

```

If you are creating a system for use in an environment that is not closed, then setting the files to world readable might not be desirable.

### 5.3. Initial Testing

At this point, all of the basic components are in place for use by kickstart when a boot floppy is used. Before actually proceeding with testing kickstart, it may be prudent to test the individual services one at a time. To do this, another Linux box can be used. The DHCP server was tested first to ensure that clients were able to boot and obtain an IP address that was in the correct range. You can view the IP address that the system received by typing `ifconfig` at the `$` prompt. Next, the DNS server was tested to make sure that the server name could be resolved to the correct IP address. You can test DNS by using `nslookup` followed by one of your host's names. Once it was determined that name resolution was working properly, the client's ability to mount the NFS directory was tested. This was tested by typing `mount -t NFS dev dir` where "dev" is the point you want to mount and "dir" is the local location you want to use for accessing the mount. After testing each of the server services successfully, these could then be eliminated as being sources of problems that might occur when trying to use the boot disk with the kickstart server.

Using the disk and `ks.cfg` file that had been created earlier, kickstart was then tested with the boot disk. When the system started, a boot prompt was presented. At this prompt, `linux ks=floppy` was entered so that the installer would know to look at the floppy drive for the kickstart configuration file. Once this was done, it was a matter of watching the install process walk through the kickstart file to configure the system and install the selected packages.

How rigorously you test may be determined by your experience setting up these services, the purpose of your kickstart server, or other means. In general, testing is recommended in a similar manner to that outlined above. This can help to isolate and eliminate any problems early.

### 5.4. PXE Boot

Now that you have the basic kickstart server working, it is time to implement the necessary services to support clients that boot using PXE. There are two items of importance. The first is that a PXE server is not actually needed. The DHCP server, along with a TFTP server, will handle PXE requests. Secondly, trying to implement the PXE server while DHCP is operational results in problems with the DHCP server. To implement the TFTP server, you will need to find an install package. Searching the Red Hat CDs, you will find that `tftp-server 0.28 - 2` is included. Install the tftp server, then edit `/etc/xinetd.d/tftp` to enable the tftp service and to update the startup location so that it will match the location indicated in your `exports` file. If you would prefer to use a different tftp server, you may be able to find one that you like at [www.rpmfind.net](http://www.rpmfind.net).

This completes the setup of the various services needed for a PXE based kickstart system. There are still a couple of things needed before the installation will function. First, although the client can get a DHCP address and find the name of the file that it needs to retrieve from the tftp server, this file is not available. As such, you will need to copy it from `/usr/lib/syslinux` into `/usr/RedHat`. `Vmlinuz` and `initrd.img` will also need to be copied from

`/usr/RedHat/image/pxeboot` to `/usr/RedHat`. The last step in setting the system up for PXE support is to create a folder under `/usr/RedHat`, called `pxelinux.cfg`, and provide the file that will inform the clients of the parameters the installation will be using. There is a system for determining the name of each potential file in this folder, but that is beyond the scope of the discussion here. For basic purposes, it is sufficient to create one file named `default`. The contents of an example file are listed below:

```
default linux
serial 0,38400n8
label linux
kernel vmlinuz
append load_ramdisk =1 initrd=initrd.img
ks=nfs=ks.kicklab.net:/usr/RedHat/ks.cfg
```

Finally, the server has all of the necessary files in place, as well as the necessary services, to support kickstart installations to a PXE client.

## 6. Final Testing, with PXE

To test the server, a Dell Optiplex GX1 PIII was first used for the client. This system was able to obtain an IP address from the DHCP server, get `pxelinux.0` from the TFTP server, find the file `default`, find `ks.cfg`, resolve the host name and complete the install using the Red Hat CD files that had been copied to the hard drive. The tests were also completed successfully using a Dell Optiplex GX240 that matched the server.

A third test system, a Dell Optiplex GX1 PII, was able to obtain an IP address from the DHCP server, but was not able to successfully complete the transfer from the tftp server. The older NIC on this system had the 3Com PXE version 0.99c.03, as opposed to the newer GX1 that had 0.99j.02. Apparently this older version has some type of compatibility issue that prevented it from working as expected. A BIOS update that corrected this problem was found at Dell's company website. *Warning - if you are using older systems, the PXE version they use may or may not be compatible with your server.*

## Final Results/Recommendations

This How-To guide, as with many similar documents, walks its way through successful scenarios while minimizing mention of the problems that were faced during the document creation. It is possible that you may encounter some problems that were experienced during the compilation of this document. A few of the problems are mentioned below. One of these problems was the "kernel panic: no init found" message which appeared on the client. This occurred early in the process while using Red Hat 7.2. No real cause was found for this problem, but reinstalling the server with Red Hat 8 eliminated the issue. Another issue dealt with attempting to use the boot floppy that had been created under 7.2 with the 8.0 server. This combination caused the client to fail with a variety of errors on the client, several of which referenced `/usr/bin/anaconda`. Simply creating a boot disk as described early in this HOW-TO document corrected this problem. The final problem worth noting related to the format of the file `dhcpd.conf`. At one point, the file `pxelinux.0` was not being properly passed to the client. Though no reason was able to be determined, it was discovered that moving the filename line to the very end of the file eliminated the problem.

**Product:** Coda File System

**Product Type:** Linux Red Hat 8.0 File System

**Author:** Sai Divvala

---

### **Problem Background**

Coda is a distributed file system that was developed at Carnegie-Mellon University. Coda is a variant of the AFS distributed file system. Coda file system project is focused on specific distributed file system functionality required for mobile computing, such as support for disconnected operation. "Disconnected operation" means that system that is a part of a networked is used without being connected to a network.

Coda provides a number of features that make it an excellent, high-performance distributed file system. Besides mobile and disconnected operation, one of Coda's important features is its extensive use of caching which means that copies of files or portions of file retrieved from Coda servers are preserved on Coda clients as long as they can be verified to match the original data stored on the Coda server. This is known as "client-side caching". This feature reduces the amount of restarting time of a Coda client by minimizing the amount of data that needs to be transmitted over the network.

Distributed file systems give access to a greater amount of storage and they can be accessed from any authorized workstation. Coda provides a single way of accessing the distributed file system that is the same on all computer systems. The entire Coda distributed file system on any workstation is mounted under the directory `/coda` which is referred to as a "global name space" because it means that all Coda systems in each administrative domain share the exact same view of the Coda distributed file system. Users can therefore locate familiar files and directories that are stored in Coda can be accessed from any machine through the same pathname. Another advantage of Coda is that it provides file and directory permissions known as Access Control Lists (ACLs). With ACLs, user can set explicit permissions for any number of individuals or existing groups. User can also create his own groups, add users to them, and then grant special sets of permissions to the group as a whole.

### **Product Placement**

As coda supports disconnected operation, it can be used in mobile applications where the disconnected operation occurs frequently. Coda file system eases the job of system administrators in maintaining FTP mirror sites. For example, `ftp.redhat.com` has many mirror sites. A Perl script is executed at each mirror site to fetch the changes that are made at RedHat site and fetches all the changed packages including unnecessary packages. If mirror sites are made as coda clients with coda server at RedHat main site, then server notifies the mirror sites (coda clients) about the updated packages. Mirror sites fetch the updated package only when someone requests for that package.

Also making WWW replication servers as coda clients avoids the manual copying of files to the servers. NFS proved to be inefficient while updating files and thus the system administrators have to upload the files manually. If the replication servers are made as coda clients, and then

they will hold the data in the cache which means that the access to data is at local disk speeds and thus improves the performance.

### **Installation Overview**

Coda architecture consists of three types of machines. They are clients, servers and system control machine (SCM). Clients are generally single-user workstations and perform operations on shared data. Clients have access to shared information, once they were authorized by the server. Servers store the shared information and provide access to clients with the shared information. SCM provides single point of control for the ease of system administration. Generally servers acts as single point of control machines.

### **Coda Terminology**

A single name space: All of the Coda files reside under single directory `/coda` which is similar to AFS file system. Coda does not have different exports. Under `/coda` all the volumes of files exported by the servers of the coda cell are visible. Coda automatically finds servers and clients need to know only name of the bootstrap server that gives the client information about how to find the root volume of coda.

Coda cell: a coda cell is a group of servers sharing one set of configuration databases. A cell may consist of one server or hundreds of servers. Out of all the servers, there will be one server that acts as single point control machine (SCM). Modifications on the configuration database are made on SCM and then the changes are propagated to the remaining servers.

Coda Volumes: Coda volume is smaller than partition and larger than a directory. Each volume has a directory tree of files. Each volume is “coda mounted” somewhere under `/coda` and forms the subtree of `/coda`. Volumes contain mount points of other volumes. Coda mount point contains information enough for the client to find the servers which stores the files in the volumes. The group of servers serving a volume is called volume storage group of the volume.

Data Storage: Coda servers store files identified by a number in a directory tree in `/vicepa`. The meta data (owners, access control lists, version vectors) is stored in an RVM data file which would often be a raw disk partition. Coda uses the meta data to support server replication and disconnected operation.

RVM: Recoverable Virtual Memory (RVM) is a transaction based library to make part of a virtual address space of a process persistent on disk and commit changes to this memory atomically to persistent storage. Coda uses RVM to manage its metadata. This data is stored in an RVM data file which is mapped into memory upon startup. Modifications are made in VM and also written to the RVM LOG file upon committing a transaction. The LOG file contains committed data that has not yet been incorporated into the data file on disk.

Client Data: Meta data in RVM (typically in `/usr/coda/DATA`) and cached files are stored by number under `/usr/coda/venus.cache`. The cache on a client is persistent. This cache contains copies of files on the server. The cache allows for quicker access to data for the client and allows for access to files when the client is not connected to the server.

When Coda detects that a server is reachable again it will validate cached data before using it to make sure the cached data is the latest version of the file. Coda compares cached version stamps associated with each object, with version stamps held by the server.

### Installation Process

First I checked if support for Coda is compiled into the kernel of my system (Red Hat 8.0) by examining the file `/proc/filesystems`. I couldn't find an entry for Coda. Then I logged in as super user and loaded the Coda loadable kernel module using the command `insmod coda`. Then I downloaded the coda software from [ftp.coda.cs.cmu.edu](http://ftp.coda.cs.cmu.edu).

### Installation of Coda Server

I copied the following files into my previously created directory "codal":

- Coda's Light-Weight Process library--Light-Weight Processes are an implementation of threads used by Coda to insure portability across a wide range of different types of systems. (`lwp-1.8-1.i386.rpm`)
- Coda's Remote Procedure Call library--Coda's remote procedure call implementation provides the basic features required for communicating and exchanging data between Coda clients and servers (`rpc2-1.13-1.i386.rpm`)
- Coda's Recoverable Virtual Memory library--Coda caches remote directory information in virtual memory which it then stored on the client systems in order to help minimize restart times and avoid redundant lookups if nothing has changed. (`rvm-1.6-1.i386.rpm`)
- Tools for Recoverable Virtual Memory Management-- Utilities used by a Coda server to cache and manage recoverable virtual memory. (`rvm-tools-1.6-1.i386.rpm`)
- Coda server--Daemons, configuration files, and startup files for a Coda server and related processes (`coda-debug-server-5.3.19-1.i386.rpm`)

Then I installed the following packages with rpm command:

```
rpm -U lwp-1.8-1.i386.rpm
rpm -U rpc2-1.13-1.i386.rpm
rpm -U rvm-1.6-1.i386.rpm
rpm -U rvm-tools-1.6-1.i386.rpm
rpm -U coda-debug-server-5.3.19-1.i386.rpm
```

After installing the packages, I have to run the shell script provided by the coda server package. Before running the script I formatted and mounted a secondary hard disk which will be used by the coda to store the data. I made a partition on the disk with ext2 file system.

```
fdisk /dev/hdb
mkfs -t ext2 /dev/hdb
```

I created a directory saying `/vicepa` and got it mounted on `/dev/hdb` which will represent the distributed file system data partition.

```
mount -t ext2 /dev/hdb /vicepa
```

Then I created two files in the `/usr/coda/logs` directory. One is used by coda to store the log information and the other one is used to log metadata. Then I started the script:

```

#vice-setup
Welcome to the Coda Server Setup script!

Setting up config files for a coda server.
Do you want the file /etc/coda/server.conf created? [yes] yes
What is the root directory for your coda server(s)? [/vice] /vice
Setting up /vice.
Directories under /vice are set up.

Is this the master server, aka the SCM machine? (y/n) y

Setting up tokens for authentication.
The following token must be identical on all servers.
Enter a random token for update authentication : foobar
The following token must be identical on all servers.
Enter a random token for auth2 authentication : foobar
The following token must be identical on all servers.
Enter a random token for volutil authentication : foobar
tokens done!
Setting up the file list for update client
Filelist for update ready.
/etc/services already has new services registered! Good.
/etc/services ready for Coda
Now installing files specific to the SCM...

Setting up servers file.
Enter an id for the SCM server. (hostname distfs)
The serverid is a unique number between 0 and 255.
You should avoid 0, 127, and 255.
serverid: 1
done!
Initializing the VSGDB to contain the SCM as E0000100
/vice/db/VSGDB set up

Setting up ROOTVOLUME file
Enter the name of the rootvolume (< 32 chars) : codaroot
Setting up users and groups for Coda

You need to give me a uid (not 0) and username (not root)
for a Coda System:Administrator member on this server,
(sort of a Coda super user)

Enter the uid of this user: 666
Enter the username of this user: codaroot
Going to rebuild the protection databases
moving /vice/db/prot_users.db to /vice/db/prot_users.db.old
moving /vice/db/prot_index.db to /vice/db/prot_index.db.old
An initial administrative user codaroot (id 666)
with Coda password changeme now exists.
A server needs a small log disk partition, preferably on a disk by
itself. It also needs a metadata partition of approx 4% of your
filesystem.

For trial purposes you may give ordinary files instead of raw
partitions. Keep all size small if you do this.
Production servers will want partitions for speed.

```

```

-----
WARNING: you are going to play with your partitions now.
verify all answers you give.
-----

```

```

WARNING: these choices are not easy to change once you are up and
running.

```

```

Are you ready to set up RVM? [yes/no] yes

```

```

What is your log partition? /usr/coda/logs/log_disk_file.log

```

```

The log size must be smaller than you log partition. We
recommend not more than 30M log size, and 2M is a good choice.
What is your log size? (enter as e.g. '2M') 2M

```

```

What is your data partition (or file)?
/usr/coda/logs/data_disk_file.log

```

```

The data size must be approx 4% of you server file space. We
have templates for servers of approx: 500M, 1G, 2.2G, 3.3G, 8G
(you can store less, but not more on such servers).
The corresponding data sizes are 22M, 44M, 90M, 130M, 316M.
Pick one of the defaults, otherwise I will bail out

```

```

What is the size of you data partition (or file)
[22M, 44M, 90M, 130M, 200M, 316M]: 22M

```

```

-----
WARNING: DATA and LOG partitions are about to be wiped.

```

```

log area: /usr/coda/logs/log_disk_file.log, size 2M.
data area: /usr/coda/logs/data_disk_file.log, size 22M.

```

```

Proceed, and wipe out old data? [y/n] y

```

```

LOG file has been initialized!

```

```

Rdsinit will initialize data and log.
This takes a while.
rvm_initialize succeeded.
Going to initialize data file to zero, could take awhile.
done.
rds_zap_heap completed successfully.
rvm_terminate succeeded.

```

```

RVM setup is done!
Your server directories will hold the files (not directories).
You can currently only have one directory per disk partition.

```

```

Where shall we store your file data [/vicepa]? /vicepa
Shall I set up a vicetab entry for /vicepa (y/n) y
Select the maximum number of files for the server.
[256K, 1M, 2M, 16M]: 16M

```

Server directory /vicepa is set up!  
 Congratulations: your configuration is ready...and now  
 to get going do the following:

```

start the auth2 server as: auth2
start rpc2portmap as: rpc2portmap
start updatesrv as: updatesrv
start updateclnt as: updateclnt -h wwcoda1
start the fileserver: startserver &
wait until the server is up: tail -f /vice/srv/SrvLog
create your root volume: createvol_rep codaroot E0000100 /vicepa
setup a client: venus-setup wwcoda1 20000
start venus: venus
enjoy Coda.
for more information see http://www.coda.cs.cmu.edu.

```

After configuring the server, I started the server using the following commands:

```

/usr/sbin/auth2
/usr/sbin/rpc2portmap
/usr/sbin/updatesrv
/usr/sbin/updateclnt -h wwcoda1
Creating /vice/spool
startserver &
[2] 3467

```

After getting the coda server running, I created the initial root volume with the following command:

```

createvol_rep codaroot E0000100 /vicepa
Getting initial version of /vice/vol/BigVolumeList.
V_BindToServer: binding to host wwcoda1
GetVolumeList finished successfully
V_BindToServer: binding to host wwcoda1
Servers are (distfs )
HexGroupId is 7f000000
creating volume codaroot.0 on distfs (partition /vicepa)
V_BindToServer: binding to host wwcoda1
V_BindToServer: binding to host wwcoda1
Set Log parameters
Fetching volume lists from servers:
V_BindToServer: binding to host wwcoda1
GetVolumeList finished successfully
distfs - success
V_BindToServer: binding to host wwcoda1
VLDB completed.
<echo codaroot 7f000000          1 1000001 0 0 0 0 0 0 0 E0000100 >>
/vice/db/VRList.new>
V_BindToServer: binding to host wwcoda1
VRDB completed.
Do you wish this volume to be Backed Up (y/n)? [n] n

```

### Installation of Coda Client

At this point I am done with the server part. Then I started to configure coda client machine. I copied the client software file into the temporary directory and installed the packages using the rpm command:

```
rpm -U lwp-1.8-1.i386.rpm
rpm -U rpc2-1.13-1.i386.rpm
rpm -U rvm-1.6-1.i386.rpm
rpm -U coda-debug-client-5.3.19-1.i386.rpm
```

After installing the files, I executed the shell script provided by the client package to start coda's cache manager(venus) and mount the coda file system:

```
/etc/rc.d/init.d/venus.init start

Starting venus: done.
Date: wed 03/05/2002

00:16:23 /usr/coda/LOG size is 239376 bytes
00:16:23 /usr/coda/DATA size is 2193368 bytes
00:16:23 Loading RVM data
00:16:23 Last init was wed Mar  5 16:23:50 2002
00:16:23 Last shutdown was clean
00:16:23 starting VDB scan

00:16:23          2 volume replicas
00:16:23          1 replicated volumes
00:16:23          0 CML entries allocated
00:16:23          0 CML entries on free-list
00:16:23 starting FSDB scan (833, 20000) (25, 75, 4)
00:16:23          781 cache files in table (6824 blocks)
00:16:23          52 cache files on free-list
00:16:23 starting HDB scan
00:16:23          3 hdb entries in table
00:16:23          0 hdb entries on free-list

00:16:23 Getting Root Volume information...
clog codaroot
username: codaroot
Password:
16:30:15 root acquiring Coda tokens!
```

### Lessons Learned/Problems

The main problem in installing coda file system is configuring the coda server. I encountered problem which displays “cannot bind to rpc2portmap” while running the vice-setup script. This problem occurs because of not running the services rpc2portmap and updateserv. Also these services should be included in the /etc/services file. Then I did the corrections to the /etc/services file and started the services rpc2portmap and updateserv which solved the problem.

Another problem I encountered is that client couldn't find the root volume which is located on the coda server. This occurs if both client and server are not in the same domain. Also we have to make sure that vice-setup script suits our coda cell. In vice-setup, coda server is pointed to Carnegie-Mellon University coda server. We have to change the name to our coda server name. Also the `/etc/hosts` file on both client and server should consist of both server name and client name with proper ip addresses.

### **Final Results/Recommendations**

Finally after installing the coda file system, I could add users and login using the "clog" command. In the future coda cell can be used for research oriented projects like disconnected operation specific projects and performance evaluation on network traffic projects (Eg. Comparative study of NFS network traffic and Coda network traffic). The coda file system is not worthwhile in implementing the ETSU university labs, as there were security concerns that need to address and also as it was a young project, there is no sufficient technical support.

### **References**

"Coda File System". Carnegie Mellon University. 30 April 2003 <[www.coda.cs.cmu.edu](http://www.coda.cs.cmu.edu)>  
Kaplan, Liliya. "The Coda Distributed File System". 30 April 2003  
<<http://piglet.uccs.edu/~cs522/proj2000/lkaplan.doc>>

**Project:** Configure X-windows for OpenBSD

**Author:** Adam Berry

---

### **Problem Background**

OpenBSD is a multi-platform 4.4BSD-based UNIX-like operating system. The goals of the OpenBSD project are focused on security, portability, and correctness. OpenBSD is widely regarded as one of the most secure Operating Systems available today. OpenBSD is designed to run on various architectures (i386, sparc, etc..) and support most software developed for UNIX environments.

While moderately difficult to install, OpenBSD lacks a decent default configurator for the X-Windows system. The X-windows system provides an alternative to a command line interface by providing a user with a functional GUI.

### **Project Goals**

The main goal of this project is to provide a template of a working XF86Config file so that others may mirror what I have discovered. The configuration tools supplied in OpenBSD can often put you on the right track, but I have yet to see a working configuration straight from the tools.

This project has no doubt been done before. The supplied file is no doubt tied to a specific set of hardware. However, a known working file can be a great reference to a beginner. I have also trimmed the file, leaving only relevant sections.

Upon successful completion of the project, users that are intimidated by all the apparent options available will be presented with a basic file with unnecessary lines.

### **Project Details**

The first task of the project was to successfully install OpenBSD. I have a dual-boot setup with Red Hat 9 so I already had this accomplished. Appendix A explains how to obtain and make a bootable OpenBSD 3.2 install CD.

After the installation is complete, it will walk you through an X configurator. Go through the motions, but don't expect it to work. The file it generates will most likely need some fine tuning.

Since I was dual-booting with Red Hat 9, and already X-windows configured on that, I had a good starting place. For most beginners, you may find it easier to install RedHat or Mandrake to start and look at the XF86Config file located in /etc/X11/ and familiarize yourself with it, or even print it out, as I did.

There are 7 main sections to the `XF86Config` file:

1. Modules – which X modules to load
2. Files – gives the search path of font libraries
3. InputDevices – specify keyboard and mouse
4. Monitor – give info on monitor
5. VideoCard – configure your video card
6. Screens – specify which devices to use (VidCard, keyboard, mouse, monitor)
7. ServerLayout – Specify default screen

The next sections are laid out in the following manner; I will present first the section of the file generated by OpenBSD's `xf86cfg` utility, then the same from Red Hat 9, then finally, a working version for OpenBSD.

Throughout the file, you will see various instances of "Identifier". It is best to think of it as a variable, it can be anything. However, as you will notice from the OpenBSD utility versus RH9's utility, a consistent naming convention can make the file easier to read and understand.

### Modules

From OpenBSD `xf86cfg` utility (bad):

```
Section "Module"

# This loads the DBE extension module.

    Load            "dbe"      # Double buffer extension

# This loads the miscellaneous extensions module, and disables
# initialization of the XFree86-DGA extension within that module.
    SubSection      "extmod"
        Option      "omit xfree86-dga"    # don't initialize the DGA
extension
    EndSubSection

# This loads the Type1 and FreeType font modules
    Load            "type1"
    Load            "freetype"

# This loads the GLX module
#    Load            "glx"

EndSection
```

From Red Hat 9 xf86cfg utility (working):

```
Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
    Load "freetype"
    Load "type1"
    Load "dri"
EndSection
```

OpenBSD Working File:

```
Section "Module"

# This loads the DBE extension module.

    Load            "dbe"      # Double buffer extension

# LINES ADDED BY ADAM

    #Load "fbdevhw"
    Load "record"
    Load "dri"

# END LINES ADDED BY ADAM

    SubSection "extmod"
        Option "omit xfree86-dga" # don't initialise the DGA
extension
    EndSubSection

# This loads the Type1 and FreeType font modules
    Load            "type1"
    Load            "freetype"

# This loads the GLX module
#    Load            "glx"

EndSection
```

The first thing I did was add all the modules listed in the RH9 file to the OpenBSD file. If this does not work right away, start by commenting out the modules that are throwing errors. In my case, the only module I had a problem with was “fbdevhw”. The module “DRI” is only needed if you want to utilize 3D acceleration.

## Files

From OpenBSD xf86cfg utility (bad):

```
Section "Files"
  RgbPath  "/usr/X11R6/lib/X11/rgb"
  FontPath  "/usr/X11R6/lib/X11/fonts/local/"
  FontPath  "/usr/X11R6/lib/X11/fonts/misc/"
  FontPath  "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
  FontPath  "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
  FontPath  "/usr/X11R6/lib/X11/fonts/Type1/"
  FontPath  "/usr/X11R6/lib/X11/fonts/Speedo/"
  FontPath  "/usr/X11R6/lib/X11/fonts/75dpi/"
  FontPath  "/usr/X11R6/lib/X11/fonts/100dpi/"
EndSection
```

From Red Hat 9 xf86cfg utility (working):

```
Section "Files"
  RgbPath  "/usr/X11R6/lib/X11/rgb"
  FontPath  "unix/:7100"
EndSection
```

OpenBSD Working File:

```
Section "Files"

  RgbPath  "/usr/X11R6/lib/X11/rgb"

  FontPath  "/usr/X11R6/lib/X11/fonts/local/"
  FontPath  "/usr/X11R6/lib/X11/fonts/misc/"
  FontPath  "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
  FontPath  "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
  FontPath  "/usr/X11R6/lib/X11/fonts/Type1/"
  FontPath  "/usr/X11R6/lib/X11/fonts/Speedo/"
  FontPath  "/usr/X11R6/lib/X11/fonts/75dpi/"
  FontPath  "/usr/X11R6/lib/X11/fonts/100dpi/"
# LINES ADDED BY ADAM
  FontPath  "unix/:7100"
# END LINES ADDED BY ADAM
EndSection
```

This section specifies the path to the RGB database and various font paths. There is usually no need to change the defaults here, but I added the UNIX font family line from the RH9 config file and no errors were thrown.

## Input Devices

Input Devices are very self explanatory, much like the rest of the file. They consist of a keyboard, mouse, and any other input device you may wish to attach.

From OpenBSD xf86cfg utility (bad):

```

Section "InputDevice"
    Identifier    "Keyboard1"
    Driver       "Keyboard"
    Option "XkbRules"    "xfree86"
    Option "XkbModel"   "pc105"
    Option "XkbLayout"  "us"
EndSection

Section "InputDevice"
    Identifier    "Mouse1"
    Driver       "mouse"
    Option "Protocol"    "PS/2"
    Option "Device"     "/dev/wsmouse"
EndSection

```

From Red Hat 9 xf86cfg utility (working):

```

Section "InputDevice"
    Identifier    "Keyboard0"
    Driver       "keyboard"
    Option      "XkbRules" "xfree86"
    Option      "XkbModel" "pc105"
    Option      "XkbLayout" "us"
EndSection

Section "InputDevice"
    Identifier    "Mouse0"
    Driver       "mouse"
    Option      "Protocol" "IMPS/2"
    Option      "Device"   "/dev/psaux"
    Option      "ZAxisMapping" "4 5"
    Option      "Emulate3Buttons" "no"
EndSection

```

OpenBSD Working File:

```

## KEYBOARD ##

Section "InputDevice"
    Identifier "Keyboard0"
    Driver "keyboard"
    Option "XkbRules"    "xfree86"
    Option "XkbModel"   "pc105"
    Option "XkbLayout"  "us"
EndSection

## END KEYBOARD ##

## MOUSE ##

Section "InputDevice"
    Identifier    "Mouse0"
    Driver       "mouse"

```

```

Option "Protocol"          "wsmouse"
Option "Device"            "/dev/wsmouse"
Option "ZAxisMapping"      "4 5"
Option "Emulate3Buttons"   "no"
EndSection

## END MOUSE ##

```

The OpenBSD `xf86cfg` utility generated a workable section, but I chose to rename the Identifier to start counting at 0 (Mouse0 instead of Mouse1), in case I wanted multiple mice present. ZaxisMapping and Emulate3Buttons are necessary only if you have a wheel mouse or a true three button mouse. Pay particular attention to the way OpenBSD names the devices. For instance, in RH9 the mouse device is `/dev/psaux`, but in OpenBSD it is labeled `/dev/wsmouse`.

### Monitor

A monitor must have an Identifier, Horizontal Sync rate, and Vertical Refresh rate. Consult your manual for the exact ranges.

From OpenBSD `xf86cfg` utility (bad):

```

Section "Monitor"
    Identifier "My Monitor"
    HorizSync  31.5 - 79.0
    VertRefresh 50-70
EndSection

```

From Red Hat 9 `xf86cfg` utility (working):

```

Section "Monitor"
    Identifier "Monitor0"
    VendorName "NEC"
    ModelName  "NEC LCD1700V"
    DisplaySize 330 270
    HorizSync  31.5 - 79.0
    VertRefresh 56.0 - 75.0
    Option      "dpms"
EndSection

```

OpenBSD Working File: same as RH9

This section shows one of the major flaws in the OpenBSD `xf86cfg` utility. There is no set naming convention to its identifiers. You may have noticed that Red Hat identifies its devices in a logical manner; Keyboard0, Mouse0, Monitor0, etc.. I found it much easier to follow that pattern than to simply create my own. Additional information is included such as model name, vendor, and display size. I also enabled the “dpms” option which adds support for VESA’s display power management system.

## Video Card

The video card section just needs an identifier and driver. You can specify the amount of RAM the card has if you choose in Kilobytes( MB \* 1024).

From OpenBSD xf86cfg utility (bad):

```
Section "Device"
  Identifier "My Video Card"
  Driver     "vga"
  #VideoRam  65563
EndSection
```

From Red Hat 9 xf86cfg utility (working):

```
Section "Device"
  Identifier "Videocard0"
  Driver     "radeon"
  VendorName "ATI Technologies" # not necessary
  BoardName  "ATI Radeon 7000"  # not necessary
  #VideoRam  65536 # not necessary, ok to specify
EndSection
```

OpenBSD Working File: same as RH9

Note that the VendorName and BoardName are not necessary, but they may be probed and automatically added. Also, the OpenBSD xf86cfg did not correctly probe the card and identified it as a generic vga.

## Screen

The screen section references the monitor and graphics card, the sets the default display options. Think of them as output devices.

From OpenBSD xf86cfg utility (bad):

```
Section "Screen"
  Identifier "Screen 1"
  Device     "My Video Card"
  Monitor    "My Monitor"
  DefaultDepth 24
  Subsection "Display"
    Depth     24
    Modes     "640x480" "800x600" "1024x768" "1280x1024"
    ViewPort  0 0
  EndSubsection
EndSection
```

From Red Hat 9 xf86cfg utility (working):

```
Section "Screen"
  Identifier "Screen0"
  Device     "Videocard0"
  Monitor    "Monitor0"
  DefaultDepth 24
```

```

Subsection "Display"
    Depth      24
    Modes      "1024x768" "800x600" "640x480"
EndSubsection
EndSection

```

OpenBSD Working File: same as RH9

To stay consistent with the naming convention thus far, I changed the Identifier to Screen0. I also changed the device and monitor to reflect there new Identifiers. The view port line was removed from the subsection because we will specify that in the next section.

### Server Layout

The server layout brings together the input and output devices. It needs an identifier, a screen to reference from the previous section, and some input devices (keyboard and mouse).

From OpenBSD xf86cfg utility (bad):

```

Section "ServerLayout"
    Identifier "Simple Layout"
    Screen "Screen 1"
    InputDevice "Mouse1" "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"
EndSection

```

From Red Hat 9 xf86cfg utility (working):

```

Section "ServerLayout"
    Identifier "Default Layout"
    Screen 0 "Screen 0" 0 0
    InputDevice "Mouse0" "CorePointer"
    InputDevice "Keyboard0" "CoreKeyboard"
EndSection

```

OpenBSD Working File: same as RH9

The only real difference was specifying the view port (the trailing zeros after "Screen 0". The OpenBSD xf86config utility wanted to specify this in the screen section, which is perfectly fine. I chose to do it here because I am running several instances of the X-windows system and want my XF86Config files to be as similar as possible across the board. It will work either way.

Everything else is optional. Once you feel comfortable with what these seven sections are doing, reboot into OpenBSD and took a look at XF86Config file with an editor. I mirrored my Red Hat XF86Config to the letter, which proved to be both good and bad.

I then tried to start x via `startx` from the command line. The screen flickered for a few moments and then shot a few errors. In this case, all errors referencing line numbers were syntax errors. The others errors were problems loading modules and a misconfigured mouse. The syntax errors were easily fixed and I moved on the next error.

The next module error was could not find module “fbdevhw”. I first commented out the Load Module “fbdevhw” to see if it gave me an error. After restarting x, the error went away. The only error left was the mouse device not found.

The error told that `/dev/psaux` could not be found. From previous experience, I was well aware that the \*BSD’s label their devices differently the Linux, so I assumed this was the problem. After `cd`’ing into the `/dev` directory, I searched for `psaux`. It was not there, so I then looked for anything that could be a mouse. I found an entry “`wsmouse`”, which if you followed above, was correctly identified by the OpenBSD `xf86cfg` utility. I changed the protocol and device to `wsmouse` and `/dev/wsmouse` respectively and continued.

After another `startx` command, the screen went blank, then X fired up.

### **Lessons Learned**

I noticed that Red Hat had little to no extra text, and the meat of the file pertained to relevant options. Conversely, the OpenBSD generated `XF86Config` was over 500 lines long. This made it very intimidating as the resulting relevant text is just shy of 200 lines.

When it comes down to it, the Red Hat `XF86Config` was an invaluable reference. From it I was able to determine what was doing what and tailor it to OpenBSD. I must admit at the projects onset that I was intimidated still by OpenBSD. Not having a GUI interface makes me a little uneasy. Another advantage of using the Red Hat file was its consistent naming convention. Red Hat labeled the devices the same way and that made it easier to determine what each section was doing. OpenBSD wanted to label everything different, which can confuse someone trying to figure out what is going on. It was hard to tell if the name was a variable, an option that needs to be set, and so forth.

On the whole, I am looking forward to configuring KDE next to get more out of my box. Learning how to configure a GUI from the command line has taught me a great deal about moving around through the command line.

### **Final Result/Recommendation**

In the end, X was successfully running on my OpenBSD 3.2 box. However, it is easy to see why KDE and GNOME have become so popular. X-windows really doesn’t add much, but it mostly used as a platform to build off of for programs such as KDE and GNOME.

If you plan on running a BSD variant and running X, take the time to look at a working `XF86Config` file from a different OS before diving right in. A working file is always the best reference.

## Appendix A: Obtaining OpenBSD 3.2

OpenBSD is freely available from <ftp://ftp.openbsd.org>. A list of mirror sites can also be found at the same location. The first complaint I have is that there are no ISO images to download. This is obviously to help promote retail sales to fund the project.

I started by mirroring all files in the `pub/OpenBSD/3.2/i386` directory. Once I had all the files downloaded, I needed to get them onto a cd. I found a windows binary for the cdrtools package and downloaded that as well.

CdrTools for Windows

<http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html>

After extracting the cdrtools package onto my XP machine, I then made a bootable ISO image using mkisofs.

### Creating a Bootable CD

```
c:\cdrtools> mkisofs
-v
-r
-T
-J
-V "OpenBSD-Current"
-b C:/OpenBSD/2.8/i386/cdrom28.fs
-c boot.catalog
-o C:/OpenBSD/OpenBSD-Current.iso
-x C:/OpenBSD/OpenBSD-Current.iso
C:/OpenBSD/.
```

Summary of options:

- -v : volume label
- -b : boot image
- -o : output file
- -x : exclude file (don't include output file in iso)
- C:/OpenBSD/. : tells where files are

### Installation

Now that I have a bootable cd, just restart and let the CD do its thing. The installation is not very friendly. There are no options to go back. After accepting the defaults for a few options, it dumps you off at a prompt, but tells you nothing.

Referring to the official installation guide, it says you will be presented with an interactive prompt. After reading some more, I figured out you must now configure the disk.

OpenBSD requires two partitions, a root and a swap before it will continue with the installation. I should also note that OpenBSD can only be installed on the first 8GB of any hard drive. This would have been helpful when to know at first when trying to dual boot with Linux.

After the disk is configured it takes about five minutes from start to finish for the full install.

## Appendix B: XF86Config File Generated During Red Hat 9 Install

```
# XFree86 4 configuration created by pyxf86config
```

```
Section "ServerLayout"
    Identifier      "Default Layout"
    Screen         0  "Screen0" 0 0
    InputDevice    "Mouse0" "CorePointer"
    InputDevice    "Keyboard0" "CoreKeyboard"
    InputDevice    "DevInputMice" "AlwaysCore"
EndSection
```

```
Section "Files"
    RgbPath        "/usr/X11R6/lib/X11/rgb"
    FontPath       "unix/:7100"
EndSection
```

```
Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
    Load "freetype"
    Load "type1"
    Load "dri"
EndSection
```

```
Section "InputDevice"
    Identifier     "Keyboard0"
    Driver         "keyboard"
    Option         "XkbRules" "xfree86"
    Option         "XkbModel" "pc105"
    Option         "XkbLayout" "us"
EndSection
```

```
Section "InputDevice"
    Identifier     "Mouse0"
    Driver         "mouse"
    Option         "Protocol" "IMPS/2"
    Option         "Device" "/dev/psaux"
    Option         "ZAxisMapping" "4 5"
    Option         "Emulate3Buttons" "no"
EndSection
```

```
Section "Monitor"
    Identifier     "Monitor0"
    VendorName     "Monitor Vendor"
    ModelName     "NEC LCD1700V"
    DisplaySize   330 270
    HorizSync     31.0 - 80.0
    VertRefresh   56.0 - 75.0
    Option        "dpms"
EndSection
```

```
Section "Device"
    Identifier     "Videocard0"
```

```
Driver      "radeon"
VendorName  "Videocard vendor"
BoardName   "ATI Radeon 7000"
VideoRam    32768
EndSection

Section "Screen"
Identifier  "Screen0"
Device     "Videocard0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
    Depth    24
    Modes    "1024x768" "800x600" "640x480"
EndSubSection
EndSection

Section "DRI"
Group      0
Mode       0666
EndSection
```

## Appendix C: XF86Config Working File

```

#
*****
# Module section -- this section is used to specify
# which dynamically loadable modules to load.
#
*****
#

Section "Module"

# This loads the DBE extension module.

    Load      "dbe"      # Double buffer extension

# LINES ADDED BY ADAM

    Load "fbdevhw"
    Load "record"
    Load "dri"

# END LINES ADDED BY ADAM

    SubSection "extmod"
        Option "omit xfree86-dga" # don't initialise the DGA
extension
    EndSubSection

# This loads the Type1 and FreeType font modules
    Load      "type1"
    Load      "freetype"

# This loads the GLX module
#    Load      "glx"

EndSection

*****
# Files section. This allows default font and rgb paths to be set
*****

Section "Files"

    RgbPath "/usr/X11R6/lib/X11/rgb"
    FontPath "/usr/X11R6/lib/X11/fonts/local/"
    FontPath "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
    FontPath "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath "/usr/X11R6/lib/X11/fonts/100dpi/"
# LINES ADDED BY ADAM
    FontPath "unix/:7100"
# END LINES ADDED BY ADAM
EndSection

```

```

#*****
# Input devices
#*****

## KEYBOARD ##

Section "InputDevice"
    Identifier "Keyboard0"
    Driver "keyboard"
    Option "XkbRules"      "xfree86"
    Option "XkbModel"     "pc105"
    Option "XkbLayout"    "us"
EndSection

## END KEYBOARD ##

## MOUSE ##

Section "InputDevice"
    Identifier "Mouse0"
    Driver "mouse"
    Option "Protocol"      "wsmouse"
    Option "Device"        "/dev/wsmouse"
    Option "ZAxisMapping"  "4 5"
    Option "Emulate3Buttons" "no"
EndSection

## END MOUSE ##

#*****
# Monitor section
#*****

Section "Monitor"
    Identifier "Monitor0"
    VendorName "Monitor Vendor"
    ModelName "NEC LCD1700V"
    DisplaySize 330 270
    HorizSync 31.5 - 79.0
    VertRefresh 56.0 - 75.0
    Option "dpms"
EndSection

#*****
# Graphics device section
#*****

## ATI RADEON 7000 ##

Section "Device"
    Identifier "Videocard0"
    Driver "radeon"

```

```

VendorName "Videocard Vendor"
BoardName  "ATI Radeon 7000"
#VideoRam  65536
EndSection

## END ATI RADEON 7000 ##

#*****
# Screen sections
#*****

Section "Screen"
    Identifier "Screen0"
    Device     "Videocard0"
    Monitor    "Monitor0"
    DefaultDepth 24
    Subsection "Display"
        Depth      24
        Modes       "1024x768" "800x600" "640x480"
    EndSubsection
EndSection

#*****
# ServerLayout sections.
#*****

Section "ServerLayout"
    Identifier "Default Layout"
    Screen 0   "Screen 0" 0 0
    InputDevice "Mouse0" "CorePointer"
    InputDevice "Keyboard0" "CoreKeyboard"
EndSection

Section "DRI"
    Group 0
    Mode 0666
EndSection

```

## Appendix D: XF86Config File Generated by Wizard

```

#*****
# Module section -- this section is used to specify
# which dynamically loadable modules to load.
#*****

Section "Module"

# This loads the DBE extension module.

    Load            "dbe"      # Double buffer extension

# This loads the miscellaneous extensions module, and disables
# initialisation of the XFree86-DGA extension within that module.
    SubSection      "extmod"
        Option       "omit xfree86-dga"    # don't initialise the DGA
extension
    EndSubSection

# This loads the Type1 and FreeType font modules
    Load            "type1"
    Load            "freetype"

# This loads the GLX module
#    Load           "glx"

EndSection

#*****
# Files section. This allows default font and rgb paths to be set
#*****

Section "Files"
    RgbPath         "/usr/X11R6/lib/X11/rgb"
    FontPath        "/usr/X11R6/lib/X11/fonts/local/"
    FontPath        "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath        "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
    FontPath        "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
    FontPath        "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath        "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath        "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath        "/usr/X11R6/lib/X11/fonts/100dpi/"
EndSection

#*****
# Input devices
#*****

Section "InputDevice"
    Identifier      "Keyboard1"
    Driver          "Keyboard"
    Option          "XkbRules"    "xfree86"
    Option          "XkbModel"    "pc105"

```

```

    Option "XkbLayout" "us"
EndSection

Section "InputDevice"
    Identifier "Mouse1"
    Driver "mouse"
    Option "Protocol" "PS/2"
    Option "Device" "/dev/wsmouse"
EndSection

#*****
# Monitor section
#*****

Section "Monitor"
    Identifier "My Monitor"
    HorizSync 31.5 - 79.0
    VertRefresh 50-70
EndSection

#*****
# Graphics device section
#*****

# Standard VGA Device:

Section "Device"
    Identifier "Standard VGA"
    VendorName "Unknown"
    BoardName "Unknown"
    Driver "vga"
EndSection

# Device configured by xf86config:

Section "Device"
    Identifier "My Video Card"
    Driver "vga"
    #VideoRam 65563
EndSection

#*****
# Screen sections
#*****

Section "Screen"
    Identifier "Screen 1"
    Device "My Video Card"
    Monitor "My Monitor"
    DefaultDepth 24
    Subsection "Display"
        Depth 24
        Modes "640x480" "800x600" "1024x768" "1280x1024"
    EndSubsection
EndSection

```

```
        ViewPort      0 0
    EndSubsection
EndSection

#*****
# ServerLayout sections.
#*****

Section "ServerLayout"
    Identifier "Simple Layout"
    Screen "Screen 1"
    InputDevice "Mouse1" "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"
EndSection
```

**Product:** FreeBSD 5.0

**Product Type:** Operating System

**Author:** Amanda Hickman

---

### **Problem Background**

FreeBSD is an alternative to the various flavors of Linux as well as the Microsoft Windows operating system. I chose to perform a FreeBSD installation in order to compare it to Linux. I wanted to determine if it is a quality product that would be useful as an alternative to Linux.

### **Product Placement**

FreeBSD is a UNIX-like operating system based on the University of California-Berkley's 4.4BSD-Lite, 4.4BSD-Lite2, and 386BSD. FreeBSD is an open source project, developed and maintained by various individuals. FreeBSD is compatible with several platforms including i386, IA-64, PC-98, Alpha/AXP, and UltraSPARC. Many companies like Yahoo, Apache, and Sony Japan use it for their daily operations. FreeBSD is comparable to OpenBSD and commercial distributions of Linux such as Red Hat and Mandrake.

### **Installation Overview**

The instructions for installing FreeBSD may be found to the FreeBSD website, <http://www.freebsd.org>.

There are several ways one can obtain a copy of FreeBSD. It is available for FTP download from several mirror sites around the world, anonymous CVS, AFS, rsync, and for purchase from online retailers. I chose to download the latest release, FreeBSD 5.0, from one of the mirrors in the United States. There are three ISO images to download and then transfer to cds. The download time was relatively short, compared with downloading Red Hat Linux 8.0 about a month ago. FreeBSD took about 45 minutes overall to download, while Red Hat took 2-3 hours for its three ISO images.

According to the FreeBSD website, the installation is supposed to be easy. The web site says "FreeBSD can be installed from a variety of media including CD-ROM, DVD-ROM, floppy disk, magnetic tape, an MS-DOS partition, or if you have a network connection, you can install it *directly* over anonymous FTP or NFS. All you need is a pair of blank, 1.44MB floppies," and to follow the directions given on the website.

I originally tried to install FreeBSD on my Pentium 150 MHz laptop with a 1.2 GB hard drive and 32 MB of RAM. Since the laptop will not boot from a cd, I had to make boot disks, which were easy to make. I followed the directions in the FreeBSD handbook. I thought the installation was going well, until I put the second boot disk in and the install got a fatal error and died. This happened twice so I decided that the laptop was not going to let me install FreeBSD.

I had much better luck with the next system I attempted to install the operating system on. The system is a Pentium III 500 MHz with 128 MB of RAM. With this system, I was able to boot from the installation cds rather than using boot disks.

While the FreeBSD website says the installation is easy, I found it anything but that. I am comparing my installation of FreeBSD to an install of Red Hat Linux. It actually took me 3 installs to finally figure out what I was doing. The first time I installed, I didn't bother to download and print the user's guide. I skimmed it on the Internet and then went straight into an installation. I honestly did not know what I was doing. The directions during the install less than helpful, but I did manage to install successfully. After that install, I decided that I would print off all 600 pages of the user's guide so I could read step by step how to install. This seemed to work better, but somehow I did something wrong and the install crashed. I ended up having to install again for a third time. This time I also chose to do a standard install instead of a custom install, since it was recommended. This install was somewhat easier.

### **Lessons Learned/Problems**

The major problem I found with FreeBSD is the lack of documentation and support. You're basically on your own, it seems, when installing and configuring this operating system. As I mentioned earlier, I also encountered a lot of trouble when trying to configure XFree86.

As for lessons learned, I've come to the realization that this operating system is for hardcore computer users. It appears to be a neat operating system, but it's not as easy to pick up as others.

### **Final Results/Recommendations**

Overall, my experience with FreeBSD was not a positive one. Compared with Red Hat, I found it difficult to install and configure. I know it is popular with many Internet companies, and I'm sure that it is a great product if someone is quite experienced and really knows what they're doing. However, if I were to implement a \*NIX type system I would probably choose a product more along the lines of Red Hat Linux since it's as easy to install and configure as a Windows installation and support and documentation is readily available, which FreeBSD is lacking.

### **References**

FreeBSD Diary. DVL Software, LTD. 30 April 2003 <<http://www.freebsdjournal.org>>  
The FreeBSD Project. 30 April 2003 <<http://www.freebsd.org>>

**Product:** OpenBSD 3.2  
**Product Type:** Operating System  
**Author:** Adam Berry

---

### Problem Background

The goals of the OpenBSD project are focused on security, portability, and correctness. OpenBSD is widely regarded as one of the most secure Operating Systems available today.

### Product Placement

OpenBSD is a multi-platform 4.4 BSD-based, UNIX-like operating system. OpenBSD is designed to run on various architectures and support most software developed for UNIX environments. Version 3.2 was used for this project.

### Test Systems

	Chipset	Processor	RAM	HDD	Optical Drives
System 1	N/A	PII 200MHZ	96MB PC100	WD 1GB	4X CD
System 2	VIA Apollo Pro133T	PIII 1.13GHZ	784MB PC133	WD 40GB	4X CDRW
System 3	nForce	Athlon XP 2000+	512MB PC2100	WD 40GB	48X CDRW

### Installation Overview

OpenBSD is freely available from <ftp://ftp.openbsd.org>. A list of mirror sites can also be found at the same location. The first complaint I have is that there are no ISO images to download. This is obviously to help promote retail sales to fund the project.

I started by mirroring all files in the pub/OpenBSD/3.2/i386 directory. Once I had all the files downloaded, I needed to get them onto a CD. I found a windows binary for the cdrtools package and downloaded that as well.

CdrTools for Windows

<http://www.fokus.gmd.de/research/cc/gclone/employees/joerg.schilling/private/cdrecord.html>

After extracting the cdrtools package onto my XP machine, I then made a bootable ISO image using mkisofs.

### Creating a Bootable CD

```
c:\cdrtools> mkisofs
-v
-r
-T
-J
-v "OpenBSD-Current"
-b C:/OpenBSD/2.8/i386/cdrom28.fs
-c boot.catalog
-o C:/OpenBSD/OpenBSD-Current.iso
-x C:/OpenBSD/OpenBSD-Current.iso
C:/OpenBSD/.
```

Summary of options:

- -v : volume label
- -b : boot image
- -o : output file
- -x : exclude file (don't include output file in iso)
- C:/OpenBSD/. : tells where files are

Now that I have a bootable cd, just restart and let the CD do its thing. The installation is not very friendly. There are no options to go back. After accepting the defaults for a few options, it dumps you off at a prompt, but tells you nothing.

Referring to the official installation guide, it says you will be presented with an interactive prompt. After reading some more, I figured out you must now configure the disk.

OpenBSD requires two partitions, a root and a swap before it will continue with the installation. I should also note that OpenBSD can only be installed on the first 8GB of any hard drive. This would have been helpful when to know at first when trying to dual boot with Linux. After the disk is configured it takes about five minutes from start to finish for the full install.

### Lessons Learned/Problems

I was disheartened to that my favorite shell was not the default choice of OpenBSD. On reboot, I had no bash shell. I downloaded the bash package from the OpenBSD Ftp and added it using:

```
% pkg_add -v bash-2.05b.tgz
```

Next I added two users, adam and Jenkins to the wheel group, and disabled root login over ssh by editing the /etc/sshd\_config file by doing:

```
% vi /etc/sshd_config
```

Then, I changed PermitRootLogin entry to "no". I next tried to configure X. I tried several configurations, but was never fully able to get it functioning.

I didn't like the fact that I failed to get X running on 3 different machines. For one, I was only able to get a working OS on two machines, with the nForce machine refusing to boot. The problem there is the IDE controller. The generic controller will not work and their no support yet for the nForce IDE controller. I found it odd that the 2 older machines worked great for OpenBSD, but the newest had the most problems.

### Final Results/Recommendations

If you like command line interface, OpenBSD is a good choice. Compared to a Red Hat install, it is very difficult. I learned a lot just by installing OpenBSD, things I did not learn by simply starting Linux's Installer.

**Project:** Automating System Setup Using RIS, Kickstart, and PXE

**Author:** Robert Nielsen

---

### **Problem Background**

Previously, the servers fell short of the original “best case” goals that had been set. The “best case” had been to create a single server for installing multiple operating system versions and that was capable of using PXE for boot services. This best case scenario would provide the ability to build systems to a known state by selecting from any of several OS versions including both Windows and Linux. As often is the case for those involved with computing systems, falling short of the original goal was not acceptable. After completion of the initial document, these items were left on the “To Do” list:

- Configure DHCP so that the appropriate IP address is passed to each client
- Attempt to combine RIS and Riprep images to clone servers with IIS and other services currently not supported by Riprep.
- Relocate the `ks.cfg` file to a shared location for easy modifiability.
- Configure IIS on the Windows 2000 Server, copy Linux files from CD to `\inetpub\wwwroot\Linux` and attempt an http based install. If successful, this could eliminate the need for the Linux based server.
- Investigate Microsoft’s implementation of PXE to determine the feasibility of capturing the PXE request before RIS takes over so that the client could select from install types.

### **Project Goals**

Of the items on the “To Do” list, the last three were viewed as most important. The investigation into “Unattended”, which was being carried out by another party on campus, offered the possibility of indirectly handling the second problem on the list. The first problem seemed reasonable in scope, but needed to wait until the server was actually moved onto the production network. The last three were of a nature that could be addressed while the systems were still running on the isolated network. The last three items were also, in many ways, intertwined in nature. Because of this, it was thought that attacking these three items would be both highly valuable and of a reasonable scope.

### **Project Details**

The first attempt involved removing the `ks.cfg` file from the bootdisk and placing it onto the NFS server. The idea was to change the `syslinux.cfg` file on the bootdisk, so that it pointed to the location of the `ks.cfg` file. This would be done by changing the line which read `ks=floppy` to `ks=nfs:ks.wvlab.net:/usr/RH8/ks.cfg` instead. Once this change was made, the floppy was used to boot and the install began. Unfortunately, things did not go as planned. Instead of retrieving the `ks.cfg` file from the NFS server, the system started going through the install interactively. Checking the messages it was noted that one read “failed to mount” followed by the IP address of our NFS server and the path to the configuration file. Etherial was used to watch the transaction in the hope of pinpointing the source of the failure. This allowed each step of the interaction between the NFS server and the client to be viewed. It was found that each step of the transaction was taking place, though the client could not mount. The decision was made to go through the interactive install to see if it failed too. Interestingly, when prompted for the NFS server name and the path to the file, the interactive installation had no problem

mounting the NFS directory and retrieving the file. In retrospect, this failure proved to be the stepping point to bigger and better things.

Since the client and NFS server were having mysterious problems, the path of least resistance was taken. There was no real concern as to how the configuration file was fed to the client. The concern was only that the configuration file handling was from a central location where it could easily be managed. This led to the decision to take advantage of the fact that http was a supported protocol for retrieving the `ks.cfg` file. Since this step would also have the potential for relating to the collaborative desire to attempt a pure http based install from the Windows 2000 server, IIS was enabled and the file was placed into a web root that was created on the same drive as the RIS images. To facilitate retrieval of `ks.cfg` from the web server, `syslinux.cfg` was modified again. This time the line was changed, `ks=nfs:ks.wvlab.net:/usr/RH8/ks.cfg` to `ks=http://risserver.wvlab.net/ks.cfg`. Attempting to boot from the floppy and start the install, the potential for progress was imminent. The client started, read the `ks.cfg` file, and went on with its install. The “To Do” list could have an item removed as `ks.cfg` could now be managed from a central location.

Finding that the configuration file could be read successfully from the IIS server, it was time to attempt “Step 3”. This tied back to the original hope for an integrated installation system. It also, if successful, would allow for removal of another item from the “To Do” list. The first part of “Step 3” was to populate the folder that held the `ks.cfg` file with the contents of all our Red Hat 8 CDs. Once the copying was completed, the `ks.cfg` file was edited and the line, which read `nfs --server ks.wvlab.net --dir /usr/RH8 to url --url http://risserver.wvlab.net/`, was changed. With this change in place, the client system was again started. Unfortunately, once again, an unanticipated problem occurred. The install went further than it had with NFS, but still did not work properly. Monitoring the client, it appeared certain that `ks.cfg` had been transferred properly and that the install was attempting to use it. While attempting to transfer files, the system presented the message “HTTP – Unable to retrieve the first image file.” At this point the IIS logs were checked to make sure that the `ks.cfg` file had been retrieved. The logs clearly indicated that it had been, but also indicated that the client had attempted to get `updates.img` and had failed. `Updates.img` isn’t a required package, so this error was not pursued any further. Returning to the client system, it was necessary to determine if it had generated any other error messages. From viewing the messages, it was determined that the install had moved beyond the failure to locate `updates.img` and had attempted to transfer `netstg1.img`. Three error messages not previously noted were found. The first error message was “<6> attempt to access beyond end of device. The second read <6> 01:01: rw=0, want=8196, limit=8192”. The third message said “<2> EXT2-fs error (device ramdisk (1,1)): read\_block\_bitmap: cannot read block bitmap – block\_group=1, block\_bitmap=8195”. The last error seemed to indicate that the problem was with the ramdisk, while the second message made it clear exactly what that problem was. The configuration on the bootdisk set the ramdisk size to 8192, but the first transfer needed more space than that. The configuration was immediately edited and the ramdisk size was increased to 16384. The idea here was to increase it more than any transfer was expected to need. After restarting the client system, it stepped through the install. After a great deal of frustration, the “To Do” list then had a second item removed. At the time, it was considered the biggest accomplishment expected to occur from this project. Tackling the issue of eliminating the boot floppy was a trip into the arena of the unknown. It

was known that http installs were possible; however, it was uncertain if they would work properly with IIS. Eliminating the boot floppy, and not breaking RIS at the same time, would be the final challenge. With the ever-present challenge, “Step 4” was the next competition to tackle.

To move forward with “Step 4”, a closer look at exactly how RIS handled communications and how it selected files for transfer to the client was needed. As part of this examination, a number of the files contained in the directory structure of the RemoteInstall share were reviewed. In doing so, one of the files examined was the RIS version of the unattended.txt file for the Windows 2000 Professional Riprep image called ristndrd.sif. In this file, the text related to this image that is displayed after a PXE client has authenticated was located. This text was listed under a section marked as [OSChooser]. It was verified that each of the install images contained a .sif file with similar contents. Since each of these was located in a folder in the form /imagename/i386/templates, an attempt was made to recreate this structure. In the templates folder, a .sif file that contained only the information in the section that seemed to be involved with the display that occurred when RIS started was created. When the client system was again booted, the new entry was displayed. One of the lines in the original files pointed to a “launch file” that resided in the i386 directory of the image. Knowing that pxelinux.0 had worked as the boot file when starting from Linux, a copy of it was placed in the image root directory and the .sif file was updated so to point to it. With this in place, the client was restarted to see what happened. At the RIS startup screen, the Linux option was selected. Somewhat surprisingly, the system started, well, almost. Though pxelinux.0 had transferred properly, the system had no way of knowing what to do next. Seeing that things were acting exactly like they had under Linux, the same path that was required for setting up our Linux PXE server was chosen and a directory was created, called pxelinux.cfg. In this directory, a default file was created and edited to match the settings that had been used in the syslinux.cfg file on the floppy disk for an http install. Having had such great success, it was fully expected this would work. It did not. The default file was double-checked to make sure that it was exactly the same as the version on the floppy. It certainly appeared to be. To be certain, the file was copied from the floppy into the pxelinux.cfg directory. With this version of the file in place, the client was restarted and the Linux option was again selected. Amazingly, it worked. It suddenly became clear; Linux and Windows do not actually save text files in the same manner. Linux only ends lines with a LF, as opposed to Windows that uses CRLF. This presents a small problem for the future, as the default file and the ks.cfg file will have to be handled while remembering this difference. Still, this seems like a minor inconvenience when compared to the consolidation that had been achieved by completing “Step 4” of the integration project.

### **Final Results/Recommendations**

There were still two things to be addressed on the “To Do” list and it was reasonably thought that a third item could have been added, “Find a way to manipulate files under Windows that does not insert the unneeded CR”. It certainly appeared that the most challenging aspects of the original goal were left behind. The project began with two completely independent systems, one running RIS and the other running Kickstart. From this beginning with two fairly well known systems, a combination was made that may possibly have been a first as no previously documentation of combining these systems to this degree could be found. Regardless of whether or not this was actually a first, completing the project and overcoming the challenges along the way certainly brought a feeling of satisfaction.

**References**

Nielson, Robert and Mario Hankerson. "Automating System Setup for CSCI 3400 using RIS and Kickstart". 2003.

**Project:** RIS and Kickstart for CSCI 3400

**Author:** Robert Nielsen

---

### **Problem Background**

Lab systems are like teen hairstyles, changing regularly and often without predictable direction. Environments of this type present one of the challenges that many systems administrators face. For lab activities to be carried out in a manner conducive to learning, it is necessary for the systems to be in a known state. Expecting users to be able to keep the systems in a known state is to a degree, unreasonable. When hands on exercises are added to the equation, the task of maintaining a known state becomes even more difficult. Users trying new things will make mistakes or they will do things that could not be anticipated. As such, we must recognize the need to be able to return the system to a known state using a method that is reasonably painless. Manually installing Windows 2000 Server, Windows 2000 Pro, or Linux on many systems is not only a time consuming process, but also leaves us with the chance that simple mistakes or inconsistencies may appear on one or more of the boxes.

### **Project Goals**

If the machines are not built exactly the same, then we may be no better off than we were when the last users walked away from the systems after making changes to them. In an attempt to alleviate some of these issues, Microsoft added Remote Installation Services to Windows 2000 and Kickstart was created for automating the installation of Linux systems. Though these systems are not exactly the same, each provides us with the ability to build a system to a basic, known configuration.

### **Project Details**

To start the preparation of the Wilson-Wallis lab for automated installation of the computers used by CSCI 3400, Windows 2000 Server on a Dell Optiplex GX1 was installed. The install was done to a four Gigabyte partition that was formatted with NTFS. For networking, the system was assigned the static IP address 10.11.11.1. During the initial install, the system was placed into a Workgroup. After the install was executed, the server configuration wizards were completed. The first step here was to install Active Directory and create a domain. This domain was called wwlab.net. The next step in the basic server setup was to configure the system as a DHCP server. Here the system was assigned the IP pool 10.11.11.5 – 10.11.11.127. The final portion of the standard server configuration was to setup the system as a DNS server. AD, DHCP, and DNS are all required for RIS, though each of them could have resided on another server if had there been a need for such a configuration. At this point it was necessary to use Add/Remove Windows Components to install RIS on the server and a hard drive partition was created and formatted to hold the RIS images. RIS was then instructed to use `d:\RemoteInstall` to store images. It was also necessary to authorize RIS to respond to clients, to provide the path to the Windows 2000 Server CD, and to create a folder for the initial image under `d:\RemoteInstall`. This initial image was given the description `win2000.srv` and the wizard attempted to copy the files from the CD. This attempt failed and we were presented with the message, “This version of the product only supports Windows 2000 Professional.” The Microsoft knowledge base was consulted and it was determined that this was a known issue and that a hot-fix was available which would correct it. The hot-fix was downloaded and applied to

the server, then, the wizard was restarted. This time the files were successfully copied to the new subfolder, `win2000.srv`. At this point, the server side setup for the basic RIS installation of Windows 2000 Server was complete.

When the configuration reached this point, it was decided that the basic system should be tested before moving on with other Windows versions or attempting to create a Riprep version of Windows 2000 Server. For this testing, a Dell Optiplex GX1 was connected to the isolated network. The system's BIOS configuration was modified so that PXE was the first boot option that the system would attempt to use. As the systems booted, the option of pressing F12 to initiate the network service boot was presented. F12 was pressed and the RIS installation automatically started by presenting an authentication screen. Once validated, the install started. The only question presented during the install asked for the "Name" of the user. The generic name, WWUser, was entered and the installation continued. It was thought that the installation would move on flawlessly; however, this did not occur. It seems that when using RIS, the client being installed does not properly join the machine domain. Instead, attempting to do so resulted in the message, "Would you like to proceed for now and try joining a domain later?" This appears to be a known problem with RIS, but the fix is not available without contacting Microsoft's technical support group. Because of this, and the fact that there appears to be no actual need for the 3400 class systems to be members of a domain, it was determined that the default setting should change so that the system would join a workgroup instead of a domain. The generic user name was inserted into the install configuration so that no user intervention would be required beyond the initial authentication. Also, though IIS and the majority of the other desired services had been installed properly, some others, including the FTP service, were not. Some reading on the structure of unattended install files provided the appropriate parameters to add so that these services would be included in the installation. When the install was tested with the new settings, it completed with no error and without presenting any prompts during the installation.

Having found success with the server edition's install, the focus shifted to creating a base image for Windows 2000 Professional. This was done using the wizard in the manner previously described. Once the files were in place, the client was started to test this new image. With multiple images now in place on the server, the startup process changed somewhat. The system now presented the two images and allowed for selection of the one desired. The Windows 2000 Professional install was selected and the system went through the various stages of the installation. Once the client was in place, it was time to move to the next level.

The next level for the systems involved further automating the client build system so that all additional software needed for the class would be available as soon as the installation completed. To get to this level, the Microsoft tool, Riprep, was used. Riprep is designed to make an image of a fully installed client machine for redistribution via RIS. Before running Riprep, Netscape browser, PUTTY, Ethereal, etc. were installed. Once these had been installed, a network connection was made to the server to run `Riprep.exe`. Riprep started and a prompt for a destination folder and a description of the image occurred. The system automatically created this new folder, `3400_win2kpro_and_apps`, and after several minutes a new image was in place on the server. To test this new image, the client was again booted by PXE. Upon starting, the system presented the new install option "Win2k pro and apps for csci 3400." As in the previous

installs, a prompt occurred for a username and password before the install would copy the files to the client system. After providing the requested information, the system was again rebuilt from the RIS image, including a fresh install of Windows. After the files were in place and the system rebooted, the applications were tested and found to be working normally.

Having added another success to the project, what was assumed to be the final image configuration for the 3400 class was undertaken. To proceed, Windows 2000 Server from the RIS server was reinstalled. With the new server install in place, the software installations were executed as had been done with Windows 2000 Professional. The other server options requested for the 3400 class, including FTP, DNS, etc., were also added. A run of Riprep was attempted as had been while building the final Windows 2000 Professional Image. This was done with a degree of trepidation as there had been previous discoveries of references that indicated IIS and DNS were not supported by Riprep. Moments later the Riprep wizard generated a window stating that IIS and DNS were running, and that an image could not be generated. This seemed a bit surprising, especially since knowing that a pure RIS install can be done which includes these services. The collaborative speculation was that using Riprep to create a layered image including these services may present a security issue and that Microsoft limited Riprep's ability to generate images as a preventative measure. Though it goes beyond the scope of the project at hand, it would be interesting to take a closer look at the images created by RIS and Riprep. If the speculation is correct, it may be possible to combine files and settings from the RIS image folder with those from a Riprep image folder in such a manner that these services would install as desired. Obviously this would not be reasonable in a production environment, but it could very well be useful in the confined settings of Wilson-Wallis.

Having completed the basic Windows based images for CSCI 3400, the Linux server that had been built as a test of kickstart was revisited. It seemed best to not take the time to "re-invent the wheel" as it were by scrapping the existing system. To be certain of where the system stood, it was determined that the best route would be to setup a client and take it through the basic install that the system was already configured to complete. This install went as expected and successfully built a basic Red Hat Linux 8 desktop. Once it was verified that the system was functional, some of the basic system configuration settings such as IP address, DHCP scope, DNS, etc. were changed. This was done as part of taking "Step 1" toward the integrated installation environment that was ultimately desired to be produced. The client install was then retested to verify the appropriately entered settings. The next step in preparing the 3400 install was to run Kickstart Configurator, which generates `ks.cfg` files. It was found that though this offers the ability to have an automated build for basic `ks.cfg` files, it did not offer the ability to select individual packages, only package by group. This seems to be a shortcoming of the Kickstart Configurator in its' current form. It was known that each Linux system automatically builds a file called `anaconda-ks.cfg` as part of the system installation. As such, advantage of this feature was taken by installing a system from CD that included all of the packages that were requested for CSCI 3400. By combining this `ks.cfg` with one that had been created using the Kickstart Configurator so that disk partition information could be included, a `ks.cfg` file that had all of the required packages and the information which would allow the install to complete without user intervention was assembled. This `ks.cfg` was tested on the client system to verify that the system was built as desired. It was surprising to find that the client reported several dependencies were unresolved when it attempted to install the individual packages listed in our

`ks.cfg`. The dependent files for the programs that fell under the required packages for the class were added. Some of the items reporting missing dependencies seemed unnecessary for the classes' purposes, so those packages were simply removed from the list in `ks.cfg`. The install was again started and the system moved through the various phases of the installation. The install completed normally and all the required packages and services were available.

Knowing that the ultimate desire for the automated installation included the ability to run both servers on the same network and at the same time, it was a corollary to take "Step 2" toward an integrated installation environment by testing the systems on the same network. To pursue this goal, both DNS and DHCP on the Linux server were disabled. This was done with knowledge that the Linux automation would take a step backwards. Using PXE to attach to either server provided the ability to only attach to that server and to retrieve only the images supported by that platform. As such, this change would require that the clients use a boot floppy to connect to the server which was no longer able to process PXE requests. The process was continued so as to allow the Windows 2000 server to handle PXE requests as an incompatibility was known to exist between some Dell GX1 systems and the Linux implementation of PXE being used. Once the networks were joined, the client which had been receiving Linux images was used to complete a Windows 2000 Professional installation. Given that no substantial changes had been made to the Windows 2000 Server, no problems were expected, but it was deemed best to verify this expectation. After this install completed, the former Windows client was booted with a Linux bootdisk that had been created from `bootnet.img`. This bootdisk had the `ks.cfg` file added to it and had updates to the configuration files so that the kickstart installation was the default selection. Once the system booted from the disk, enter was pressed and the installation proceeded.

### **Final Results/Recommendations**

This final success marked the end of the basic configuration for RIS and Kickstart as required by CSCI 3400. The initial overall desire was not completely fulfilled as it had been hoped to build every system to completion and to eliminate any need for a bootdisk. These steps would at this point fall into the "To Do" list for Wilson-Wallis. More completely, the "To Do" list might read as follows:

- Configure DHCP so that the appropriate IP address is passed to each client
- Attempt to combine RIS and Riprep images to clone servers with IIS and other services currently not supported by Riprep.
- Relocate the `ks.cfg` file to a shared location for easy modifiability.
- Configure IIS on the Windows 2000 Server, copy Linux files from CD to `\inetpub\wwwroot\Linux` and attempt an http based install. If successful, this could eliminate the need for the Linux based server.
- Investigate Microsoft's implementation of PXE to determine the feasibility of capturing the PXE request before RIS takes over so that we could allow the client to select from install types.

**References**

"Remote Operating System Installation Overview". Microsoft Corporation. 28 October 1999.  
30 April 2003 <[http://www.microsoft.com/windows2000/techinfo/howitworks/  
management/remotever.asp](http://www.microsoft.com/windows2000/techinfo/howitworks/management/remotever.asp)>

Red Hat Documentation. 30 April 2003 <[http://www.redhat.com/docs/manuals/linux/RHL-7.3-  
Manual/custom-guide/ch-kickstart2.html](http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/custom-guide/ch-kickstart2.html)>

**Project:** RIS-Kickstart for CSCI 4417

**Authors:** Amanda Hickman, Trey Buck, and Adam Berry

---

### **Problem Background**

The CSCI 4417/5417 class at East Tennessee State University deals with several operating systems throughout the semester. The students learn to install and configure Windows 2000 Professional, Windows 2000 Server, Windows XP Professional, and the latest version of Red Hat Linux. The lab monitor, in the lab where the students work, is responsible for restoring all computers to a certain state. The lab monitor needs a simple method of installing any of the operating systems the students use in a quick manner, which requires little attention. The solution is to use Windows Remote Installation Services (RIS) and Linux's Kickstart to provide a method of unattended installation for all operating systems needed.

### **Project Goals**

The goal of this project is to successfully install Windows and Linux remotely using one machine. This project is mostly for the CSCI 4417 class, however we believe that other universities that teach with multiple platforms would benefit from our project.

### **Project Details**

RIS is a component of Windows 2000 Server, which allows for the remote installation of Windows operating systems to client computers. The RIS server must be a Windows 2000 Server machine in which all hard drives have been NTFS formatted and a separate disk partition has been created for the RIS images. Additionally, the following services must be configured on the network as well: DHCP, DNS, and Active Directory. The Appendix at the end of this document is a "How-To" guide for RIS and also details the requirements for the RIS clients and server.

RIS was originally developed for the remote deployment of Windows 2000 Professional only. With the release of Service Pack 3, additional functionality has been included to allow the deployment of Windows 2000 Server and Windows XP Professional. Microsoft support article 308508 details how to create a Windows 2000 Server image for RIS. Microsoft support article 304314 details how to create a Windows XP image for RIS deployment. Also, with Windows XP requiring the product activation key for an installation to continue without assistance, the key must be added to the .sif file (the image configuration file) for the XP image.

Once these images have been created, the client machines are able to perform a PXE boot, which enables them to connect to the RIS server. If a machine is not able to boot via PXE, then a RIS installation disk is necessary. After the machines have booted, the user is presented a menu to select the appropriate image to install. The user must then enter an administrator username and password combination in order to install the image. When the machine reboots, the image is copied onto the client machine's hard drive.

## Linux Kickstart

In addition to automating the installation of various versions of Windows, it was necessary to provide a mechanism to automate the installation of a Linux distribution as well. We chose to use Red Hat Linux, as it was the distribution previously implemented in the lab. We did, however, upgrade to version 8.0, which at the time was the most recent.

We wanted both Windows and Linux to be installed from the same physical server to eliminate redundancy. Since a RIS install requires the server to be running Windows 2000 Server, we needed to decide how to install Linux from a Windows machine. We ultimately decided to do the installation via FTP. The choice between an FTP and an HTTP install was rather arbitrary. There seem to be no real advantages on way or the other, but we suspected there would be fewer bugs in an FTP install. An NFS install would have been ideal as the Kickstart file could be retrieved over the network. Unfortunately, Microsoft does not include an NFS daemon with its operating systems. It must be noted that there are commercially available NFS daemons and clients, but having no budget for this project effectively eliminated this option for us.

Setting up the FTP server is not complex, but there is little documentation on how the install tree should be configured. For clarity it is noted here. An install tree is the end result of merging the files from all the distribution's CD-ROMs under one directory structure. To create an install tree, choose the location you would like to house the installation files and copy the entire contents of the first CD-ROM to that location. Merge the contents of each subsequent CD-ROM into the existing directory structure. For example, the first and second CD-ROMs of the Red Hat distribution each contain an RPMS directory. The files contained in the RPMS directory of the second CD-ROM should be copied into the RPMS directory created when the first CD-ROM was copied to the hard drive.

Being able to install Linux via network is indeed a convenience, but we also needed a mechanism to automate the process and standardize the installation. Fortunately, Linux provides Kickstart functionality, which makes it relatively simple to automate and standardize the installation process. Installation settings can be specified in the `ks.cfg` file. The `ks.cfg` file essentially tells the installer how to set up the system.

We specified the appropriate settings in the `ks.cfg` file. Any existing partitions were set to be removed and the Master Boot Record cleared. This was to ensure we were starting the installation with the system in a known state. Due to the variety of hardware in the lab, it would be unrealistic to hard code some settings into the `ks.cfg` file. Whenever possible, we chose options which gave us more flexibility. Swap files were set to their recommended sizes. Primary partitions were configured to grow to fill the disk. Mice, monitors, and graphics cards are all auto probed by the installer. This gave us the flexibility to install onto any hardware configuration that may appear.

The final step in automating the Linux installation process is getting the installation to start. This turned out to be a simple matter of modifying the `syslinux.cfg` file to automatically pass the "ks" option to the kernel, thus initiating an installation. Our `syslinux.cfg` file contains the following entry:

```
default ks
prompt 0
timeout 10
label ks
kernel vmlinuz
append ks=floppy initrd=initrd.img
```

Linux can now be successfully installed by ensuring the FTP server is running, inserting the floppy disk, and booting the machine. The administrator password will be set, the selected packages installed, and X Windows configured.

### **Lessons Learned/Problems**

We had very few problems with this project. The documentation for both RIS and Linux Kickstart is thorough. The Microsoft Support page was also helpful in providing a solution to installing both Windows 2000 Server and Windows XP Pro using RIS.

As for lessons learned, we learned that it is possible to use RIS and Linux Kickstart together. Our solution was somewhat different than the solution provided by Mario Hankerson and Robert Nielsen, however we still accomplished the same task.

### **Final Results/Recommendations**

We have provided a RIS how-to at the end of this document that gives step-by-step instructions for RIS (see appendix A).

### **References**

"How to Deploy Windows XP Images from Windows 2000 RIS Servers". Microsoft Corporation. 30 April 2003 <<http://support.microsoft.com/default.aspx?scid=kb;en-us;304314>>

## **Appendix A**

### Introduction to Remote Installation Services

Windows Remote Installation Services (RIS) allows remote installation of Windows 2000 professional. This guide shows the steps required for installation and configuration of RIS server as a component of Windows 2000 Server, which permits remote deployment to RIS clients. The RIS server must meet all hardware requirements which are described at the end of this document. The server's hard drives also must be NTFS formatted. Additionally, Windows 2000 server is installed and configured correctly along with the following correctly configured services, DHCP, DNS, and Active Directory. Remote Installation Services also needs to be selected from the optional components list, in order to facilitate successful deployment of RIS, which can be chosen from the initial server setup or afterwards from control panel, add/remove programs, add/remove windows components, and finally checking the option remote installation services. RIS cannot be installed on the primary system hard drive partition; therefore, a separate partition is required to install the RIS server.

### Installing Windows 2000 Remote Installation Services

1. Click the start menu and type rsetup.exe in the run dialog box.
2. Press next when you see the RIS welcome screen appear.
3. Enter the location where you want to install the RIS files and click next.
4. Select the option "respond to client computers requesting service" and click next.
5. Enter the location of the Windows 2000 Professional source files and click next.
6. Enter the directory name for files to be copied to on the RIS server, "win2000.pro", "winxp.pro" or "win2000server.pro" and click next.
7. Enter a description of the install and any help text in the boxes provided and click next on this prompted screen.
8. The setup wizard shows a summary of what will be completed and after reviewing this information press finish.
9. RIS is now installed, but we now have to authorize the RIS server, which is activated within Active Directory.

### Authorizing and Configuring RIS

1. You must be logged in as a domain administrator to authorize the DHCP server in Active Directory, so the RIS sever will be able to service client computers.
2. Press the start menu, go to programs, administrative tools and then to DHCP.
3. Select your DHCP server in the left pane, right click and go down and select authorize.
4. Press the start menu, go to programs, administrative tools, and then to Active Directory Users and Computers.
5. Right click on the domain and select delegate control.
6. The delegation wizard starts, press next.
7. Add the users you want to have permissions to do RIS installs and press next.
8. Choose the option "join a computer to the domain" under delegate common tasks and press next.
9. Press finish.
10. Go to start menu and type mmc in the run dialog box, and hit enter.
11. Add the group policy snap-in; expand local computer policy, computer configuration, windows settings, security settings, local policies, and finally user rights assignment.

12. Double-click on log on as batch job.
13. When the security settings dialog box appears press add.
14. Choose the user you want to grant permissions to perform RIS installations, press add and lastly press the ok button and close the mmc.

#### Installing Windows 2000 Professional on client computers

If a computer is able to perform a PXE boot then it is not necessary to create a boot disk. The computer's BIOS must be set for a PXE boot and then follow the instructions given on the computer screen. Otherwise, if a computer is not able to PXE boot, then a boot disk must be created. The following steps describe the creation of the RIS boot disk.

1. Create an RIS boot disk, by typing `\Reminst\Admin\I386\Rbfg.exe` in the run dialog box and press the ok button.
2. The Windows 2000 Remote Boot Disk dialog appears insert a floppy disk and press create disk.
3. Go to the client computer and insert the floppy disk and start the computer.
4. Press F12 when prompted for RIS boot.
5. Press Enter
6. Enter the domain where the RIS files are located on the RIS server and enter a valid domain username and password.
7. Press Enter
8. The installation of Windows 2000 Professional begins on the client computer.

#### Configuration Options

The above steps describe a simple RIS setup. RIS may be configured to completely automate the entire setup process. This is done by editing the .sif file found in the i386/Templates directory for each image. The file may be edited to create a default username/password combination, join a domain/workgroup, or set a default screen resolution. This is only a partial list of the configuration options available.

**Appendix B**Server minimum requirements

- Pentium or Pentium II 200 MHz
- 128 MB of Ram
- 2-GB primary system partition
- 2-GB drive dedicated to the Remote Installation Server
- 1 Floppy disk for the RIS boot disk
- 10/100 mb/s network card

Client minimum requirements

- Pentium 166 MHz
- 64 MB of RAM
- 2-GB drive minimum
- 1 floppy disk for the RIS boot disk
- PXE DHCP-based boot ROM and a network card

**Note:** The information include in this document comes from:

Donald, Lisa & James Chellis. "MCSE Windows 2000 Professional Study Guide". Sybex 2000. Chapter 2 pages 41 - 95.

**Product:** Red Hat Linux 8.0

**Product Type:** Operating System

**Author:** Steve Fritts

---

### **Problem Background**

This project is primarily aimed at producing a useful exercise for the CSCI 4417 class. Past classes have included assignments for user account setup, including password and home directory setup. This is not usually sufficient for typical users, however.

### **Project Goals**

My goal is to create a document that will walk students step-by-step through creating user accounts and setting up the user environments via the command line. This document will also include an exercise for students to complete, which will allow them to get some "hands-on" experience in setting up user environments in Linux 8.0.

### **Walkthrough**

For this section, I am going to include my primer and exercise set. This document will take the students through creating user accounts and changing user environment variables.

### **User Account Setup**

The first step in being able to work in a Linux environment is to have a Linux user account created. There are several steps that must be taken in order to add a new user to the system. These include:

- Creating a user account record
- Setting the account's password
- Specifying a login shell for the account
- Creating a home directory for the account
- Populating the account's home directory with various useful files

Performing the steps above with the command line interface is straightforward. We'll start by creating a user account. The simplest way to do this is to type `useradd [username]`. For example, to create an account for a user named Kermit the Frog, I might want to use an account name of "kfrog". To create the account, then, I would type `useradd kfrog`. Note: when accounts are created this way they are locked-out by default. They will be unusable until a password is created for them, as described next. Using the `useradd` (or `adduser`) commands to create accounts automatically sets some user environment options for the accounts, as will be described later.

### **Setting Account Passwords**

To set an account's password type the command `passwd [username]`. This prompts you to enter (and reenter for verification) the new user account's password. So, for example, to set Kermit's password to "misspiggy", I would use the command `passwd kfrog`. Linux will then prompt me to enter the password twice. Once this is accomplished, the password is saved and the user can begin using it immediately.

## User Login Shells

Each user account created in Linux has a login shell. The default shell for Linux is `/bin/bash`. This can also be changed.

## User Home Directories

Each user account created has their own home directory in which they can create new directories and files. By default, Red Hat Linux 8.0 specifies `/home/[username]` as the home directory for a particular user account. Administrators can change this in a couple of ways. One way is to use the `-d` switch for `useradd`. The syntax is `useradd [username] -d [home directory]`. For example, to create a home directory other than the default for Kermit, I could use the command `useradd kfrog -d /lillyPad`.

By typing the `ls` command after running the `useradd` command above, I can see that the `lillyPad` home directory was created as shown below:

```
[root@dhcpc2 /]# ls
bin          dev          hi           lib          misc        proc        tftpboot    var
BOFResult.txt  downloads  home        lillyPad    mnt        root        tmp
boot        etc          initrd      lost+found  opt         sbin        usr
[root@dhcpc2 /]#
```

Note that the default files created for the `kfrog` account are similar to those that would be created in a default `/home` directory. Listed below are the files for Kermit's custom home directory "lillyPad", and a default home directory for user `sfritts`:

```
[root@dhcpc2 /]# ls -al /home/sfritts
total 32
drwx-----  3 sfritts  sfritts    4096 Apr 16 06:11 .
drwxr-xr-x   7 root      root      4096 Apr 16 21:09 ..
-rw-r--r--   1 sfritts  sfritts    24 Feb 17 18:24 .bash_logout
-rw-r--r--   1 sfritts  sfritts   191 Feb 17 18:24 .bash_profile
-rw-r--r--   1 sfritts  sfritts   124 Feb 17 18:24 .bashrc
-rw-r--r--   1 sfritts  sfritts   854 Feb 17 18:24 .emacs

[root@dhcpc2 /]# ls -al lillyPad
total 6
drwx-----  2 kfrog    kfrog     1024 Apr 16 22:29 .
drwxr-xr-x  23 root      root     1024 Apr 16 22:29 ..
-rw-r--r--   1 kfrog    kfrog     24 Apr 16 22:29 .bash_logout
-rw-r--r--   1 kfrog    kfrog    191 Apr 16 22:29 .bash_profile
-rw-r--r--   1 kfrog    kfrog    124 Apr 16 22:29 .bashrc
-rw-r--r--   1 kfrog    kfrog    854 Apr 16 22:29 .emacs
[root@dhcpc2 /]#
```

Notice the four files shown above: `.bash_logout`, `.bash_profile`, `.bashrc`, and `.emacs`. We will come back to these later.

## Creating User Accounts Manually with the `/etc/passwd` File

The settings for the items shown thus far are saved in one file--`/etc/passwd`. Users with root privileges can edit this file and add accounts (and account settings) by hand. Here is a breakdown of accounts in the `/etc/passwd` file:

AccountName:Password:UserID:GroupID:UserDescription:HomeDirectory:DefaultShell

The first two fields are AccountName and Password. The UserID field is a unique identifier that is used to associate accounts with all of their files and directories. This identifier is actually what is used by Linux to keep track of files associated with specific users. The human-friendly account name is what is displayed to us. The GroupID field is the user account's default login group. Linux always creates this when a new user account is created. For Red Hat Linux, this default GroupID, called a *user private group*, is identical to the UserID. The next item, "UserDescription", can be used to list information about the user associated with the account. The last two items are the user's home directory and login shell. Below is a snapshot of the first set and last set of accounts that exist in my `/etc/passwd` file (accounts I have created appear at the bottom of the file):

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
sfritts:x:500:500:Steve Fritts:/home/sfritts:/bin/bash
test:x:501:501:./home/test:/bin/bash
test1:x:502:502:./home/test1:/bin/bash
beavis:x:503:503:./home/beavis:/bin/bash
monster:x:504:504:./hi:/bin/bash
kfrog:x:505:505:Kermit the Frog,Sesame Street,423-SESME-ST,423-THE-
POND:/lillyPad:/bin/bash
```

Notice that only the `sfritts` and `kfrog` accounts have user descriptions. Also notice that all password values are set to 'x'. This indicates that *shadow passwords* are in use. The x above in the password field indicates that all user account passwords are stored in a separate file from the `/etc/passwd` file. Note: the "root" account always has UserID and GroupID of "0", and the first 500 UserIDs and GroupIDs are reserved for Linux entities.

To manually add a user account, I can simply open the `/etc/passwd` file in a text editor like `vi` or `pico`, and add a new line using the fields shown above. But, I will also have to create a new home directory using the `mkdir` command, then copy certain default user files into this directory (we'll cover those files later).

### `/etc/shadow`

Here is a breakdown of the `/etc/shadow` file:

AccountName:EncryptedPassword>LastPasswordChange:DaysUntilChangeAllowed  
:DaysBeforeChangeRequired:DaysWarningBeforePasswordExpires  
:DaysBetweenExpirationAndDeactivation:AccountExpires:SpecialFlag

The fields in the `/etc/shadow` file are described below:

*AccountName* - same as in the `/etc/passwd` file

*EncryptedPassword* - this is the password associated with the account name

*LastPasswordChange* - This represents the number of days since January 1, 1970, that the last password was changed.

*DaysUntilChangeAllowed* - This represents the number of days before a password change is allowed. Setting this to "0" allows the user to change it as often as they want.

*DaysBeforeChangeRequired* - This represents the number of days before a password change is forced. If this value is set to "99999", the user will never be asked to change the password.

*DaysWarningBeforePasswordExpires* - This sets the number of days' warning before a password change is required. If a root user wanted to give other users a week's notice, they would set this field to "7".

*DaysBetweenExpirationAndDeactivation* - This represents the number of days that an account may be expired before the account is disabled.

*AccountExpires* - This shows the number of days (from January 1, 1970) that will transpire before the account expires. To turn this feature off, set this field to "-1".

*SpecialFlag* - This flag is reserved and usually remains empty.

Below is how the accounts shown above appear in the `etc/shadow` file:

```
root:$1$aajaa5AIx$zNi6i6HnEJy/NjepFRKQh1:12159:0:99999:7:::
bin:!:12159:0:99999:7:::
daemon:!:12159:0:99999:7:::
adm:!:12159:0:99999:7:::
lp:!:12159:0:99999:7:::
...

sfritts:$1$h0iYyJB0$bsUdEtq35QKTwx1EMVRR40:12100:0:99999:7:::
test:$1$z5EgJ6Lf$XyWtdo6JRzC1fLnDNutTh/:12159:0:99999:7:::
test1:!!:12159:0:99999:7:::
beavis:butthead:12159:0:99999:7:::
monster:!!:12159:0:99999:7:::
kfrog:!!:12159:0:99999:7:::
```

*Note:* *shadow passwords are enabled by default in Red Hat Linux 8.0.* To disable shadow passwords I can use the command `pwunconv`. This will take the encrypted password for each account in the `/etc/shadow` file and add it to the appropriate line in the `/etc/passwd` file. For example, if I look at my `/etc/passwd` file after running `pwunconv`, the accounts in this file now appear as follows:

```
sfritts:$1$h0iYyJB0$bsUdEtq35QKTwx1EMVRR40:500:500:Steve
    Fritts:/home/sfritts:/bin/bash
test:$1$z5EgJ6Lf$XyWtdo6JRzC1fLnDNutTh/:501:501:~/home/test:/bin/bash
test1:!!:502:502:~/home/test1:/bin/bash
beavis:butthead:503:503:~/home/beavis:/bin/bash
monster:!!:504:504:~/hi:/bin/bash
kfrog:!!:505:505:Kermit the Frog,Sesame Street,423-SESME-ST,423-THE-
    POND:/lillyPad:/bin/bash
```

Note: Running the `pwunconv` command will delete the `/etc/shadow` file altogether. To restore shadow passwords, use the command `pwconv`.

### More User Environment Settings

There are a lot of user environment settings left to examine. The next topic we will examine is the `/etc/skel` directory. This directory holds files that will be created in `/home` directories for new users. For example, if I run the `ls` command to examine the `/etc/skel` directory on my system, I see the following:

```
[root@dhcpc2 /]# ls -al /etc/skel
total 9
drwxr-xr-x  2 root    root    1024 Feb 17 18:17 .
drwxr-xr-x 54 root    root    4096 Apr 16 23:35 ..
-rw-r--r--  1 root    root      24 Aug 23  2002 .bash_logout
-rw-r--r--  1 root    root    191 Aug 23  2002 .bash_profile
-rw-r--r--  1 root    root    124 Aug 23  2002 .bashrc
-rw-r--r--  1 root    root    854 Aug 28  2002 .emacs
[root@dhcpc2 /]#
```

Notice that the four files shown earlier in the `/lillyPad` and `/home/sfritts` directories are shown here. So, if I add a file here, what happens? Let's find out ...

```
[root@dhcpc2 /]# pico /etc/skel/hello.txt

[root@dhcpc2 skel]# ls -al
total 10
drwxr-xr-x  2 root    root    1024 Apr 17 00:08 .
drwxr-xr-x 54 root    root    4096 Apr 16 23:35 ..
-rw-r--r--  1 root    root      24 Aug 23  2002 .bash_logout
-rw-r--r--  1 root    root    191 Aug 23  2002 .bash_profile
-rw-r--r--  1 root    root    124 Aug 23  2002 .bashrc
-rw-r--r--  1 root    root    854 Aug 28  2002 .emacs
-rw-r--r--  1 root    root     32 Apr 17 00:08 hello.txt
[root@dhcpc2 skel]#
```

You can see above that I have added a file to the `/etc/skel` directory. So ... if I add a new user

...

```
[root@dhcpc2 skel]# useradd grover
[root@dhcpc2 skel]# ls -al /home/grover
total 28
drwx-----  2 grover  grover  4096 Apr 17 00:10 .
drwxr-xr-x   8 root    root    4096 Apr 17 00:10 ..
-rw-r--r--   1 grover  grover   24 Apr 17 00:10 .bash_logout
-rw-r--r--   1 grover  grover  191 Apr 17 00:10 .bash_profile
-rw-r--r--   1 grover  grover  124 Apr 17 00:10 .bashrc
-rw-r--r--   1 grover  grover  854 Apr 17 00:10 .emacs
-rw-r--r--   1 grover  grover   32 Apr 17 00:10 hello.txt
[root@dhcpc2 skel]#
```

... you can see that the new user has the `hello.txt` file in their directory. However, if I look at the files in a previously-existing user ...

```
[root@dhcpc2 skel]# ls -al /lillyPad
total 6
drwx-----  2 kfrog    kfrog      1024 Apr 16 22:29 .
drwxr-xr-x  23 root      root       1024 Apr 16 22:45 ..
-rw-r--r--   1 kfrog    kfrog        24 Apr 16 22:29 .bash_logout
-rw-r--r--   1 kfrog    kfrog       191 Apr 16 22:29 .bash_profile
-rw-r--r--   1 kfrog    kfrog       124 Apr 16 22:29 .bashrc
-rw-r--r--   1 kfrog    kfrog       854 Apr 16 22:29 .emacs
[root@dhcpc2 skel]#
```

... the new file `hello.txt` isn't there. This is a limitation to be aware of. If I decided, after creating accounts, that I want a new file or program to be added to all users home directories, I will have to go back and add the file to directories of users who already exist. Adding a file to `/etc/skel` will only affect new users created after the file was added to the `/etc/skel` directory.

To make changes to the user environment that will affect all users, you must make changes to the `/etc/profile` file.

## Setup

Root users can run the setup utility to configure items such as keyboard, mouse, system services that run when Linux boots, password settings, X configuration, and other settings. For users who do not have administrator access, they will be prompted for the root password in order to run setup, as shown below:

```
[sfritts@dhcpc2 sfritts]$ setup
You are attempting to run "setup" which may benefit from administrative
privileges, but more information is needed in order to do so.
Password for root:
```

Typing in the correct root password at the prompt shown above will open up the menu shown in Figure 1. From here, users can use the arrow keys or tab keys to navigate around and make menu selections.

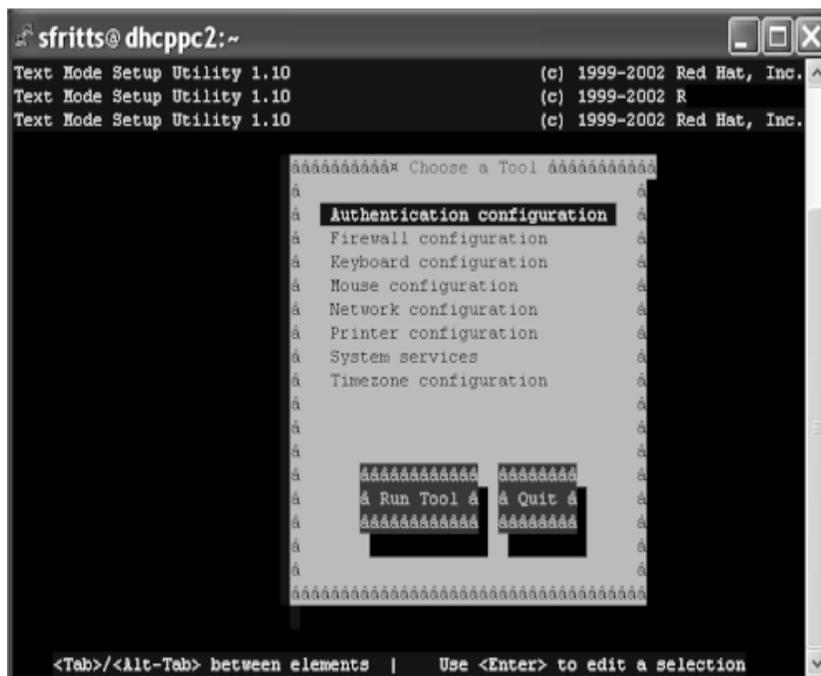


Figure 1: Red Hat Linux setup utility

## Environment Variables

Red Hat Linux includes several items called environment variables that are used to save user settings. To see a list of environment variables that are set, type `set`. On my machine I get the following list (the variables are displayed in bold font):

```
[sfritts@dhcpc2 sfritts]$ set
BASH=/bin/bash
BASH_VERSION=( [0]="2" [1]="05b" [2]="0" [3]="1" [4]="release" [5]="i686-
pclinux-gnu" )
BASH_VERSION='2.05b.0(1)-release'
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=157
DIRSTACK=( )
EUID=500
GROUPS=( )
G_BROKEN_FILENAMES=1
HISTFILE=/home/sfritts/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/sfritts
HOSTNAME=dhcpc2
HOSTTYPE=i686
IFS=$' \t\n'
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=56
LOGNAME=sfritts
LS_COLORS=no=00:fi=00:di=00
MACHTYPE=i686-pc-linux-gnu
MAIL=/var/spool/mail/sfritts
MAILCHECK=60
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/sfritts/bin
PIPESTATUS=( [0]="0" )
PPID=8387
PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME%%.*}:${PWD/#$HOME/~}\007"'
PS1='[\u@\h \W]\$ '
PS2='> '
PS4='+ '
PWD=/home/sfritts
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:interactive-
comments:monitor
SHLVL=1
SSH_CLIENT='192.168.0.2 4702 22'
SSH_TTY=/dev/pts/0
SUPPORTED=en_US.UTF-8:en_US:en
TERM=xterm
UID=500
USER=sfritts
_clear
langfile=/home/sfritts/.i18n
tm=00
```

We will now discuss a few of the more common variables here.

**HOME** - the HOME variable allows users to set which directory they start from when they login. To change it, type `export HOME=` and then the new directory path in double-quotes.

**PATH** - this variable sets the path Linux will use to search for programs or commands that you wish to run. If a command you want to run is not listed in one of the here, you must simply include the full path name. You may add as many directories here as you wish. Just type each path separated by a colon. End the string with a period--this tells Linux to include whichever directory you are currently in when searching for executables . An example would be `PATH=/usr/bin:usr/local/bin.`

**PS1** - this variable determines what the shell prompt looks like. If I type `PS1=boss`, the shell prompt will now display `boss>`.

**SHELL** - this variable allows you to change your default shell.

**LOGNAME** - use this variable to change your login name

**UID** - this shows the UserID that is associated with the user's account name. Linux uses this number to associate files with particular users.

**USER**-this shows the user account name

### Exercises

I have included here some simple exercises you can use to help you practice using some of the commands and environment variables described above.

1. Create two user accounts: "user1" and "user2". Create one of them using the `useradd` command, and another by adding a new line to the `/etc/passwd` file. After creating the accounts, log out as root and see if you can login with both accounts. Remember that user accounts are locked out when they are created. You must set the password before you will be able to login with the account.
2. Make the appropriate changes to the `etc/shadow` file so that the user accounts created in problem 1 will:
  - a. require passwords to be changed after 30 days
  - b. warn users 3 days before their passwords expire
3. Use the appropriate command to remove shadow passwords, then use a text editor to examine the `/etc/passwd` file. Then use the appropriate command to restore shadow passwords. Use a text editor to view the `/etc/passwd` and `/etc/shadow` files.
4. As system administrator, you decide that you want to welcome all new users by placing a file in their home directory that the users can read when they log in. Create a README file in Linux using a text editor, then place the file in the appropriate directory so that the file will be added to each new user's home directory when their user account is created.
5. There are several switches you can use with the `useradd` command other than the one I described above for setting the home directory. Create a new user account, using the appropriate switch to set an account expiration date of two weeks from today. After you've verified that this has been set, check the user's home directory and make sure that the README file from problem 4 is there.

6. Log in with one of your newly-created user accounts. Change the appropriate environment variable so that your username displays at the command prompt. Now change the `.bash_profile` file in your home directory so that this change will take effect every time you login.

### Exercise Solutions

1. To create "user1" I could simply type `adduser user1`. Creating the other user account can be accomplished by adding the following line to the `/etc/passwd` file. To create the second user, I would add the following line to `/etc/passwd`:

```
user2:x:550:550:User 2:/home/user2:/bin/bash
```

Remember that GroupIDs and UserIDs should be above 500. I used 550 just to be safe. Since I declared user2's home directory to be `/home/user2`, I have to create this directory manually by using the following command:

```
mkdir /home/user2
```

Next, I must copy the files from `/etc/skel` to user2's home directory with the following command:

```
cp /etc/skel/* /home/user2
```

Finally, I must set passwords for each account with the `passwd` command.

2. The `/etc/shadow` file should look similar to the following:

```
user1:!!:12159:0:30:3:::
user2:!!:12159:0:30:3:::
```

Note that fields 2 and 3 (encrypted password and date password was last changed) in your file will be different from what is shown above.

3. Simply use the `pwconv` and `pwunconv` commands to complete this.

4. I used `pico` to create the README file ...

```
[root@dhcpc2 root]# pico /etc/skel/README
```

... typed in a few lines of text ("Welcome to the company", etc.), and saved the file. Using the ls command confirms that the file has been created ...

```
[root@dhcpc2 root]# ls -al /etc/skel
total 11
drwxr-xr-x    2 root    root    1024 Apr 17 09:17 .
drwxr-xr-x   54 root    root    4096 Apr 17 06:08 ..
-rw-r--r--    1 root    root      24 Aug 23  2002 .bash_logout
-rw-r--r--    1 root    root    191 Aug 23  2002 .bash_profile
-rw-r--r--    1 root    root    124 Aug 23  2002 .bashrc
-rw-r--r--    1 root    root    854 Aug 28  2002 .emacs
-rw-r--r--    1 root    root     32 Apr 17 00:08 hello.txt
-rw-r--r--    1 root    root     55 Apr 17 09:17 README
[root@dhcpc2 root]#
```

... Now I will create my account. The switch for expiration dates is -e, so my syntax would be as follows ...

```
[root@dhcpc2 root]# useradd user3 -e 2003-05-01
useradd user3 -e 2003-05-01
[root@dhcpc2 root]#
```

Next I will check the shadow file to verify that it was added ...

```
user3:!!:12159:0:99999:7::12173:
```

Subtracting the date the password was last updated (which is when the account was created--12159) from the expiration date (12173) gives me 14 days, which is correct.

6. I used the following command to change my command prompt (results are shown below also):

```
[user1@dhcpc2 user1]$ export PS1=$USER
user1#ls -al
total 36
drwx-----    2 user1    user1    4096 Apr 17 09:31 .
drwxr-xr-x   10 root    root    4096 Apr 17 09:21 ..
-rw-----    1 user1    user1    205 Apr 17 09:44 .bash_history
-rw-r--r--    1 user1    user1     24 Apr 17 09:21 .bash_logout
-rw-r--r--    1 user1    user1    191 Apr 17 09:21 .bash_profile
-rw-r--r--    1 user1    user1    124 Apr 17 09:21 .bashrc
-rw-r--r--    1 user1    user1    854 Apr 17 09:21 .emacs
-rw-r--r--    1 user1    user1     32 Apr 17 09:21 hello.txt
-rw-r--r--    1 user1    user1     55 Apr 17 09:21 README
user1
```

Notice that I have PS1 equal to \$USER. Without quotes. This means take the value in the USER variable and set it as the prompt. I could have simply used export PS1="user1" instead. Notice that my prompt above is now simply user1, without brackets around it or anything like that. It would also be a good idea to have some sort of character at the end of my command prompt string such as "\$" or ">", so I have to include that when specifying my prompt string.

Also note that this change will only last until I log out. To make a permanent change, I must add this command to my `.bash_profile` file for user1, which looks as follows:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
PS1=$USER">"

export PATH
unset USERNAME
```

The line above in bold is what I added. Notice that I added ">" to the end of the prompt, so that my command line prompt now looks like this:

```
user1>
```

### Final Results/Recommendations

I learned quite a bit about setting up user accounts and manipulating account environment variables as a result of completing this project. I hope that these exercises will help students in the 4417 class learn to use the commands I have described. I found an excellent source for scores of How-To documents and other guides at the Linux Documentation Project website<sup>2</sup>. There were no How-To guides for what I have described here, so maybe this one can be added.

A good way to expand on what I have covered in this project would be to automate user account creation by using some of the commands described here in a script.

### References

- Stanfield, Vicki and Roderick W. Smith. *Linux System Administration*. 2001.
- "The Process of Creating User Accounts". Red Hat Linux 8.0: The Official Red Hat Linux System Administrator Primer. 15 Apr. 2003  
<<http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/admin-primer/s1-acctsgroups-proc.html>>
- Wirzenius, Lars, Joanna Oja and Stephen Stafford. "The Linux System Administrator's Guide: Version 0.7" 17 April 2003 <<http://www.tldp.org/LDP/sag/index.html>>

---

<sup>2</sup> <http://www.tldp.org/index.html>.

**Product:** Solaris 9 x86 without Sparc

**Product Type:** Operating System

**Author:** Todd Franklin

---

### **Problem Background**

The purpose of this project is to install and run Sun Solaris without a Sparc system. The only option is to use Solaris x86 (for Intel).

### **Product Placement**

Solaris is a Unix operating system that has different features than most Linux operating systems, and has a very good reputation and a large market. The newest version available is Solaris 9 and it is the only Solaris x86 version available for download from Sun's download center, so this is the version I chose to use.

### **Installation Overview**

The first step in installing Solaris 9 x86 is to obtain the software. I downloaded the software from Sun's download center for \$20. To order the CD's costs approximately \$100. The next step is checking the HCL (Hardware Compatibility List). I have found that this is often overlooked and sometimes it is just assumed that the hardware components of a PC will be acceptable to any operating system. It is true that sometimes a component that is not on the HCL will still function correctly with generic drivers. Sun had an addition to there HCL that included devices that were compatible with third party drivers. Several components were not listed in the HCL including the motherboard, a Shuttle AI61. However, none of these devices have presented any apparent problems. The BIOS on the motherboard allowed booting from the CD drive, which was necessary to perform the installation.

My installation of Sun's Solaris 9 x86 was performed on an Athlon 850 box. I used a 2 gig hard drive. Most of the components were on the compatibility list. The installation software included an Installation CD, a Software 1 CD, and a Software 2 CD. There was an additional CD available, the Languages CD. This CD was not needed for the installation. I understand that this CD simply includes additional languages. I also used a Windows 98 Start-up disk to use FDISK to delete all exiting partitions on the hard drive.

I began the installation by booting from the Installation CD. This CD is used to load the installation program, but none of the actual Solaris OS. I completed the required steps (which will be outlined below) and rebooted, only to get several errors. I then deleted the partitions again and started the process again from the beginning. I tried this unsuccessfully several times. Each time I would receive different error messages. Sometimes I would receive write error messages at the end of the Installation CD procedure. I began to think that the hard drive was bad. I then remembered hearing something about problems with the Installation CD and that the process could be done without the use of the Installation CD. I could not find any reference from Sun relating to this, but I verified this with an IT consultant and also found several other references that claimed the Installation disk was not needed and in fact would not work in most instances. I started the process over, beginning with the Software 1 CD. After deleting the existing partitions on the hard drive, I rebooted from the Software 1 CD. The PC booted to a

menu driven interface.

The first screen was the Solaris Device Configuration Assistant. The first statement explained the initial objectives as being to scan to identify system hardware, list identified devices, and boot from a specified device. The options were:

- To perform a full scan to identify all system hardware choose continue
- To diagnose possible scan failures, choose Specific Scan
- To add new or updated drivers choose add driver

The default legend at the bottom of the screen had “F2 continue”, “F3 Specific Scan”, “F4 Add Driver”, and “F6 Help”. Every screen in the menu driven interface had the option of F2 which would select the default action and most screens included the help option. The only time I tried the help option nothing happened. There was also a back option on many screens, and then there were specific options for the particular screen such as the Specific Scan or Add Driver option. All of the options listed by the legend at the bottom of each screen were activated by the use of a function key.

There was a statement “About Navigation”. It explained that the mouse could not be used. It also stated that if the keyboard did not have function keys or if they did not respond, then the legend could be changed to Esc keys by pressing Esc.

At next screen shown after pressing F2 (Continue) the application scanned for devices and compiled a device list. The list was displayed and there was an option to identify devices not on the list by choosing “Device Tasks”. The devices tasks screen had the options of changing devices or accepting the selections. The next screen loaded the drivers and then the Boot Solaris screen appeared.

This screen was a little confusing. Installation guides (Not Sun’s) also note this confusion. This screen prompts the user to “Select one of the identified devices to boot the Solaris Kernel”. The devices shown were the hard drive and the CD-ROM. It would seem that this is asking for the device from which the operating system will be booted from, but this is not the case. What is really wanted is the device from which to boot the Kernel from during the installation. There is also a Boot Tasks option which leads to options for viewing/editing autoboot settings, viewing/editing property settings. and setting the network configuration.

The autoboot settings included current boot device, autoboot time-out in seconds (default was 5 seconds), and turning autoboot on or off. The property settings included the output device (screen) and input device (keyboard). The network configuration strategy was to choose either DHCP or RARP. The only changes I made was to increase the autoboot time-out to 20 seconds and to set the network configuration to DHCP. (Although the network setting was of little consequence, since I was installing this on a Windows network.) I have since read that there are potential problems in changing any network settings here and that this should be done after the install.

From here the interface changed to a command line interface. A choice was given for the type of installation - Solaris Interactive (1) or Custom Jumpstart (2). After typing 1 and pressing enter, a short time passed while the system was being configured. Then a language and local had to be entered.

Once again a menu interface was used, showing the “Solaris Installation Program”. This screen gave an explanation of the tasks that would be accomplished.

1. Identify peripheral devices
2. Identify your system
3. Install the Solaris software

Then a series of screens for Kdmconfig were used to view and edit the properties for the “Window System Configuration”. This included the video/monitor, keyboard and pointing device. Then the configuration could be tested (or bypassed). The testing of the windowing configuration simply displayed a screen with a grid of colored ovals. In each oval was the name of the color. A mouse pointer became present to click the appropriate space to indicate that the configuration was functional.

From here to the end of the installation, the interface consisted of two windows. A window in the upper left-hand corner labeled “Solaris Install Console” gave an ongoing summary of configuration files being installed or scripts being ran. The window in the center of the screen, labeled “Sysidtool” (and later “Suninstall”), was the interactive interface for completing the installation.

The next steps included entering and verifying the hostname, entering the region, country and time zone and entering the time and date.

The next step was choosing whether to do a standard install which consisted of installing “from a standard Solaris distribution” or a flash install which consisted of installing “from one or more flash archives.”

The next choice was to select the software from:	
Entire Distribution plus OEM Support	1985MB
Entire Distribution	1984MB
Developer System Support	1797MB
End User System Support	1394MB
Core System Support	644MB

I selected the entire distribution(without OEM Support).

Then came a series of screens concerning disk selection and partition creation, and file system and disk layout. I choose the automatic layout and then accepted the default file system and disk layout. A final question was if a remote file system was to be mounted (I chose no) and then a summary of the options chosen (since the window system began) were displayed and they could either be accepted or changed. After I accepted the settings the system rebooted, the actual installation of the software began.

It took another 15 or 20 minutes to finish installing from disk 1 and the system rebooted again. I was then prompted for a root password. After entering the password the rest of the installation did not require any further input from me, but it did take a considerable amount of time to install the software from disk two. The system finally rebooted and brought up the login box. The installation process took almost 2 hours, most of which occurred after all of the user input had been completed. This does not include the failed attempts using the Installation disk.

### **Lessons Learned/Problems**

There are several areas concerning this installation which need more discussion. One is that the installation from the Software 1 disk was very similar to the installation from the Installation disk up to the first reboot, but not exactly the same. This could be due to the fact that I chose different options sometimes, just to view what was available and it then change the order of a couple of screens. It should also be pointed out that if I chose to bypass the testing of the window system then the interface would go to the command line mode and not enter the window interface. The command line mode seemed a little faster, but I did not stay in the command line mode when I used the Software 1 disk.

This being said, I have to wonder what the purpose is of the installation disk. Since I could not find a reference from Sun to the option of bypassing the installation disk, and I was not able to successfully use the installation disk, I do not know if it could offer a variation on the rest of the software installation or even make the installation proceed at a faster rate. In Windows 2000 there are several installation options and some result in a 16-bit install and others result in a much faster 32-bit install. I thought that this might be similar to the purpose of the installation disk. I am also disappointed that Sun does not address the problems with the installation disk.

Another point that needs mentioning is upon the first successful install, during which I was asked to enter the root password, I encountered an additional problem. This problem was not apparent at first, but it eventually led me to have to re-install Solaris again. This occurred when I was prompted for a root password and was instructed that if I did not want to use a root password then I should press the return key twice. This would lead someone to believe that there would be no negative impact on the actual operation of the system if a root password was not used. There would obviously be a negative impact on the security of such a system, but since this no one else has access to this system and it is simply for educational purposes, that was not a factor.

After deciding not to use a password for simplicities sake, I was able to log in to the Common Desktop Environment without a problem. However, once the GUI was loaded, I did not have access to the management features. When trying to access a component such as Users, I would be prompted to enter the password for root. Since there was not a password for root, it was impossible to enter one. When I pressed the return key, I was denied access. I do recall having the same thing happen a few years ago with an installation of Red Hat, under the same circumstances.

Beyond these issues, is the issue of the installation process itself. It seems that this installation process requires much less input than a Windows 2000 system. Although there are several options to change settings, if the machine has compatible components, then most of these options

don't need to be used. It would seem that this system could be more easily automated for multiple machines of the same build. The installation still requires a minimal amount of knowledge of certain aspects, such as disk and file layout, so it is more than just anyone could do. However, it does not require expertise beyond what most competent individuals with a decent amount of computer skills could achieve. Although I am not (at this time) a Linux/Unix person, I do realize that the simplicity of the installation is made possible by the design for much more (and simpler) user configuration after the installation than Windows platforms. Since the system doesn't define everything at installation, it is not difficult to make it what it needs to be on any particular machine.

I would further like to comment on the system itself. I spent quite a bit of time checking out different components of the GUI. I will admit that the interface had a refreshing effect, for someone who spends a lot of his time looking at Windows interfaces (98, 2000, and XP aren't very different from each other). Basically the Common Desktop Environment, as the GUI is called, consists of a panel across the bottom of the screen. There are several pull down menus that actually pull up. These menus are as follows:

- Links - includes install icon (which is included in all the menus), Web Browser, Personal Bookmarks, and Find Web Page
- Cards - includes an icon with the date (brings up a calendar for the month) and Find Cards
- Files - includes Home Folder, drive related items, Encryption, Compression, Archive and Find File
- Applications - which includes links to different applications
- Personal Printers - includes Default and Print Manager
- Tools - includes Desktop Controls, Solaris Management Console, and other tools
- Host - includes Performance Meter, This Host, System Info. Console, and Find Host
- Help - includes a variety of help related items
- Trash - includes Trash and Empty Trash Can

The panel also includes a padlock icon to lock the desktop, a globe icon that brings up a dialog box to enter the path of where you want to go, and an exit icon to log out. Finally there are four buttons numbered one through four that apparently allow you to be performing work in one layer of the desktop and then switch to another layer where you can have a whole different set of windows open. When you toggle between these you see only what was opened under that layer of the desktop. It only takes a little while of playing around to understand how to navigate around the system. It is simple and effective. I was disappointed in the computer management console and thought that it was very limited in what management could be done. I realize, of course that the power of the system lies in the use of the command line interface.

I did attempt to configure the network settings through the command line interface and through the use of the text editor. I was unsuccessful at getting the system to operate on my Windows network. When I consulted Sun's documentation, the only help I found in integrating a Solaris system into a Windows network was available in the form of a course which was available for a fee.

**Final Results/Recommendations**

I was surprised with the limited HCL that Solaris x86 has. This might be because Sun is not going all out with Solaris x86 and that they would much rather sell the standard Solaris and the hardware that it needs to run on. Although the GUI is very user friendly, it is very slow to respond. Considering the results of this installation and the overall cost of Sun's proprietary systems, if I were considering what platform to use for an organization, I would probably not choose Solaris. However, I am still working with the system and find it quite interesting.

# *Security*

**Product:** Advanced Intrusion Detection Environment (AIDE)

**Product Type:** Utility

**Author:** Gunter Wambaugh

---

### Problem Background

For an administrator, it is difficult to detect an intruder and whether or not an intruder modified critical files. AIDE is a utility to detect alterations to the local file system. As an administrator, I was interested in the usefulness of AIDE.

### Product Placement

The Advanced Intrusion Detection Environment (AIDE) is a utility that maintains a database of user-specified attributes of user-specified files. AIDE compares the attributes of each file in the database with the attributes of the corresponding file on the file system. When the attributes don't match, AIDE alerts the user. The idea is to locate possible intrusions from a cracker.

AIDE is open source and is advertised as a free replacement for the semi-free Tripwire. AIDE is easy to install and use and its possible uses are limited only by the user's creativity. AIDE is available at <http://www.cs.tut.fi/~rammer/aide.html>. It supports several platforms, including Linux and Microsoft Windows. This document describes AIDE running on Linux. In order to install AIDE on Linux, the *mhash* library must also be installed. The mhash library is available at <http://mhash.sourceforge.net>.

### Installation Overview

The following steps were taken to complete the installation (commands included):

1. Unpack mhash:  
`$tar -xzvf mhash-0.8.17.tar.gz`
2. Compile mhash:  
`$cd mhash-0.8.17`  
`$. /configure`  
`$make`
3. Install mhash:  
`$make install`
4. Unpack AIDE:  
`$tar -xzvf aide-0.9.tar.gz`
5. Compile AIDE:  
`$cd aide-0.9`  
`$. /configure`  
`$make`
6. Install AIDE:  
`$make install`

After AIDE is installed, a configuration file must be created. AIDE does not come with a default configuration file, and AIDE is mostly useless without one. The format of AIDE's configuration

file is similar to Tripwire's, making a conversion to AIDE from Tripwire easier. For detailed information on the format of the configuration file try `$man aide.conf`. AIDE will look for the configuration file in `/usr/etc` by default. This can be overridden with `$aide -config=configfile`. The following is the configuration file I used:

```
# aide.conf
#[ 02.04.03 | Gunter Wambaugh ]

# Advanced Intrusion Detection Environment configuration file.

# The database to read from.
database = file:/var/lib/aide.db

# The new database to create.
database_out = file:/var/lib/aide.db.new

# Gzip the database.
gzip_dbout = yes

# Level of verbosity.
verbose = 20

# ---- Rules for adding files to the database. ----

# Files to include.
/bin R
/boot R
/etc R
/lib R
/opt R
/sbin R
/usr R

# Files not to include.
!/dev
!/home
!/lost+found
!/mnt
!/proc
!/tmp
!/var
!/root
```

This configuration puts a gzipped database in `/var/lib`. It tells AIDE to record the permissions, inode, user, group, size, modification time, creation time, and md5 sum of all the files in `/bin`, `/boot`, `/etc`, `/lib`, `/opt`, `/sbin`, and `/usr` as specified by the default R group.

Groups are used to define what attributes AIDE should store on the specified file(s). AIDE has a list of pre-defined groups. A user can create a custom group by adding or subtracting pre-defined groups. The R group is equivalent to `p+i+n+u+g+s+m+c+md5`.

### First Run

After a configuration file is created, AIDE must create an initial database: `$aide --init`. While AIDE is running, it might be a good idea to avoid making changes to the file system. On my Celeron 366 with 160MB of RAM, `$time aide --init` reported:

```
real    12m5.513s
```

```

user      3m30.210s
sys       0m39.480s

```

After the database has been created, it has to be renamed from `aide.db.new` to `aide.db`. AIDE can now be used to detect file changes. This can be achieved by `$aide --check`. The results will be sent to stdout.

For normal operation, it is a good idea to automate the execution of AIDE. Naturally, cron is an ideal choice. The following is the cron script I created and put in `cron.weekly`:

```

#!/bin/sh

# [ Gunter | 02.04.03 ]
# aide cron script.
# This just checks the aide database, it does not create a new one.

nice aide --check &> /var/log/aide-`eval date +%m_%d_%y`.log

```

This cron script will result in a log file located in `/var/log`. Running AIDE weekly may be conservative for some file systems. I chose to run AIDE weekly for my laptop because I don't modify system files very often; however, my laptop is connected to the ETSU network several hours a week.

### Lessons Learned/Problems

I encountered compiler errors when building AIDE with gcc 2.95.3. Some of the source files had to be compiled without the `-I/usr/include` directive. I had to do the following manual compilations:

```

$gcc -DHAVE_CONFIG_H -I. -I/home/gunter/tmp/aide-0.9/src -I.. -
I/home/gunter/tmp/aide-0.9/include -static -c conf_yacc.c
$gcc -DHAVE_CONFIG_H -I. -I/home/gunter/tmp/aide-0.9/src -I.. -
I/home/gunter/tmp/aide-0.9/include -static -c commandconf.c
$gcc -DHAVE_CONFIG_H -I. -I/home/gunter/tmp/aide-0.9/src -I.. -
I/home/gunter/tmp/aide-0.9/include -static -c gen_list.c
$gcc -DHAVE_CONFIG_H -I. -I/home/gunter/tmp/aide-0.9/src -I.. -
I/home/gunter/tmp/aide-0.9/include -static -c compare_db.c

```

### Final Results/Recommendations

The primary use of AIDE is to detect intrusions. It is important to note that AIDE does not do anything to prevent or stop an intrusion. Aside from detecting intrusions, AIDE can help to recover from them. Since AIDE will indicate modified files, an administrator will know exactly what files need to be restored. Knowing that all modified files have been restored can give an administrator a peace of mind without the hassle of reinstalling the operating system.

I plan to install AIDE on machines that I am responsible for. I have had instances where I thought a person might have gotten in to the system, but was unsure if he modified anything. Now that I have AIDE, I will know.

### References

AIDE. 29 April 2003 <<http://www.cs.tut.fi/~rammer/aide.html>>

**Product:** Deep Freeze  
**Product Type:** Utility  
**Author:** Mario Hankerson

---

### **Problem Background**

This product evaluation began with the desire to find an easier and simpler way to restore computers to a known state without system downtime. Deep Freeze solves this problem by saving, or “freezing” the state of a computer system at selected intervals. By saving this information it prevents system corruption at the hands of an innocent or malicious user. Overall, this product evaluation will illustrate the usefulness and potential need for using Deep Freeze to eliminate software configuration problems, system corruption, and the loss of effective time use for system administrators.

Deep Freeze promotes itself to be a Windows protection system that “freezes” configurations in order to protect computers from the installation and deletion of programs by unauthorized persons. It works on both individual computers and networked systems. The purpose is to increase security as well as standardize computers and prevent users from wreaking intentional havoc while also stopping unintentional mistakes from causing system failure. In academic and corporate environments, installing Deep Freeze will ensure restoring computers back to known states will be seamless; thus, standardization and security will be significantly increased.

### **Product Placement**

The utility Deep Freeze, unlike imaging products on the market, returns Windows configurations files back to their default settings upon rebooting. It has two editions, standard and professional. The standard version was used for this product evaluation and did not disclose information for certain functionalities. The vendor claims the software is less restrictive than alternatives, and allows full access to users without the fear of computer downtime. Although individual computers could benefit from Deep Freeze, networked systems that offer unrestricted use to various users will reap the most benefit from this product. By having the ability to install this program and make changes to the system at once without having to visit each station, the system administrator saves valuable time. Another time saving solution lies in the basic problem fixed by this software--the permanent freezing of Windows settings upon reboot. If the configurations are unable to be altered, the system administrator wastes no time reformatting or re-imaging computers. All computers on the network function in a standardized manner as a result of using this utility.

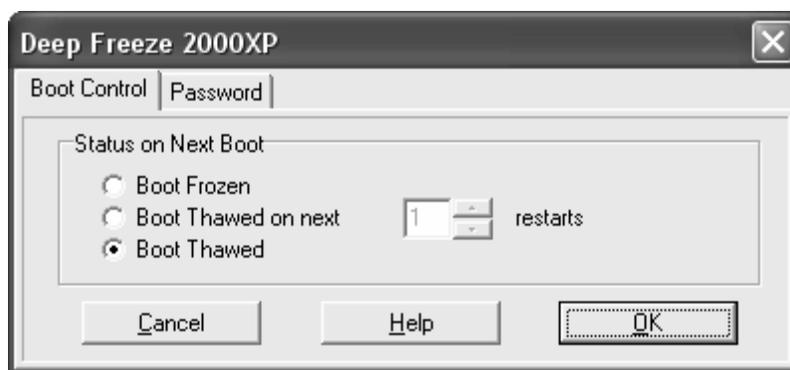


Figure 2.1: Deep Freeze 2000XP

Deep Freeze has special features that enhance it as a utility that should be adopted. Security can be added by using stealth installation, which does not display a system tray icon. The configuration of Deep Freeze can be customized through the administrator GUI so that it is only accessible by authorized system administrators. In addition, Deep Freeze allows for such things as scheduled restarts, which gives the system administrator the option to set up varying days and times for the program to initiate rebooting. This allows the system to be maintained in a standard format at optimal times. Deep Freeze also permits idle time restarts that increase system optimization for users by restoring the system to its standard state. If a computer has not been used for a designated amount of time, Deep Freeze will reboot the system. It will not reboot the system again, however, until the workstation has been used after the initial reboot. This feature keeps it from continuously rebooting if it is not being used.

One particular feature that gives the program versatility is “Thawed Space.” This allows the system administrator to designate a portion of the hard drive as “thawed” for permanent storage. The amount of space available can be as little as 16MB to as large as 2GB. What this means is that for certain instances a user will want a storage area that is permanent. One of the downfalls of a freezing program is that it allows one to easily bring back his or her system in the event of a computer malfunction. However, it will return everything to its previous state. This means that documents and saved changes made after that freezing point are no longer available to the user. This is why a thawed space is so important. It gives the user a separate space that is not touched in the event that the computer must be returned to a previous state.

The utility implements increased levels of password flexibility for greater security. The administrator has the ability to generate one time passwords as needed in order to allow student technicians, for example, to help make software changes. Deep Freeze also has the ability to assign up to four permanent passwords. Additionally, passwords are encrypted for tighter security. Security, simplicity, standardization, time management, flexibility, and accessibility make up the common threads of Deep Freeze.

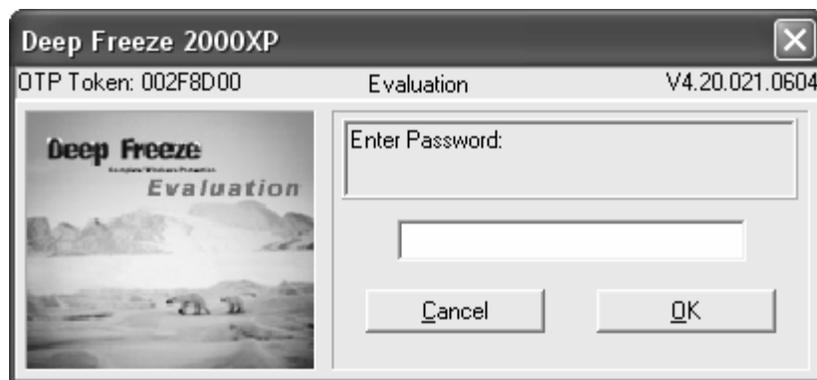


Figure 2.2: Deep Freeze Login

One feature contributing to the goal of accessibility is Scheduled Maintenance. This is made easy by the ability to choose times for the computer to reboot into an unfrozen state and be set to reboot at a later time back into its frozen configurations. This keeps the system administrator from having to process shutting down Deep Freeze every time routine maintenance is performed. This proves particularly important for updating software and the like. Deep Freeze even gives the option of locking out the keyboard and mouse during these scheduled times to prevent user interference. The Deep Freeze Command Line Control (DFC) allows the utility to be easily deployed across a network so administrators can remotely control system operational states. With all of Deep Freeze's features being modular, the administrator can pick features best suited to the system and configure them to meet the organization's specific needs.

### Core Functionality

As the product was being tested, no documentation was found that echoed how the product actually worked. The vendor claims that the software is patent-pending and presumably they do not tell how the software accomplishes the tasks because of trade secrets. However, system administrators can look at the processes running on a system with Deep Freeze and relatively determine how the utility accomplishes what the vendor claims. After extensive review and usage of the utility, one can reasonably state that Deep Freeze is a client/server application masked as a single utility. Additionally, when a person views the system processes, two new processes will be displayed: DFServeEx.exe and FrzState.exe. The assumption is that a server is running and a client is running also in a background process. What is believed to happen is that the client intercepts messages and sends them to a deep freeze server where the changes are committed. If the computer is frozen its changes do not take affect. A great comparison would be to say that Deep Freeze is like a Trojan horse, because it waits and intercepts requests based on how it is configured and does some user defined action.

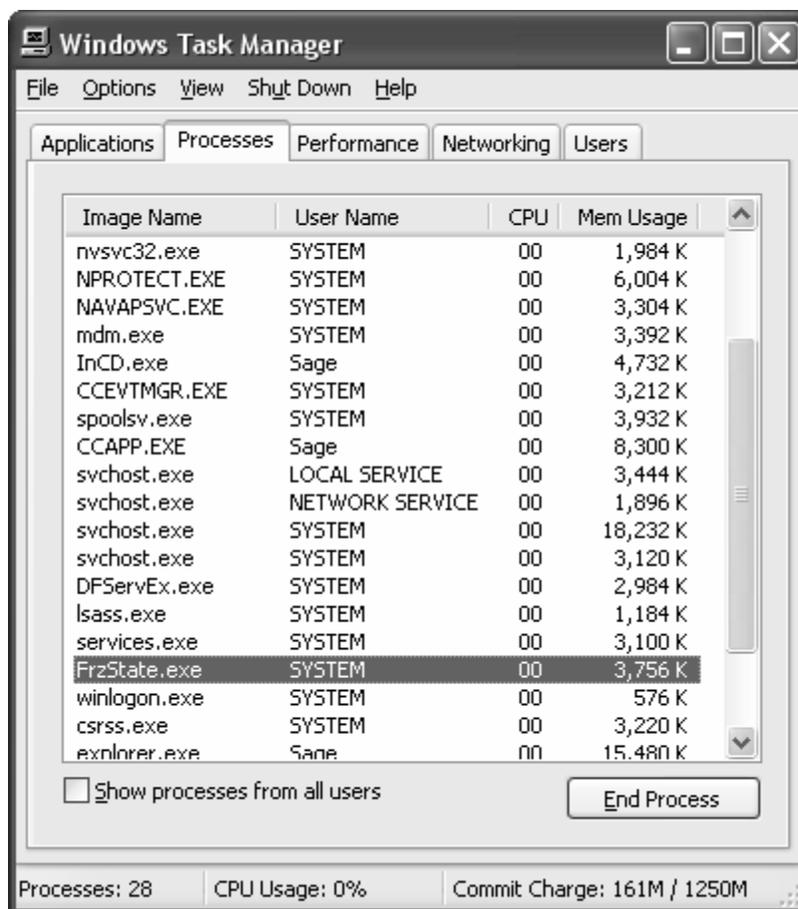


Figure 2.3: Deep Freeze processes running

### Installation Overview

The minimum system requirements necessary for installing Deep Freeze are: a Pentium processor, Windows 2000/XP, 64 MB's of memory, and 10% free hard disk space. The first step in the process which must be accomplished is to be absolutely certain that all programs are closed before beginning the Deep Freeze installation. The install process is automatic and intuitive. In addition, Deep Freeze requires virtually no setup or configuration by default other than creating an administrators password, setting up "thawed space," and selecting which drive(s) that need to be monitored. Once the software has been downloaded from the vendors' website and saved into a local folder on a system, i.e., install folder, double clicking on the icon entitled "Deep Freeze" will launch the install wizard and the user will be prompted for an install path for the utility. After the utility has been installed, the system must be restarted for the program to complete installation.

In order to gain access to the login/control screen two options exist: (1) right click on the Deep Freeze icon in the system tray or (2) use the hot key combination of Ctrl+Alt+Shift+F6. At the login/control screen the system administrator will be prompted to enter a password if he/she did not enter one during the initial setup. After the user has entered the utility password, configuration changes can be made. The user is presented with several choices, including but not limited to disabling, enabling, schedule, and etc., Deep Freeze options that will allow the

system administrator to change the default options to those that will meet his/her organizational needs.

### **Lessons Learned/Problems**

The Deep Freeze install was extremely smooth with only a minor glitch which occurred because all system programs were not stopped during installation. This resulted in Norton Antivirus Professional Edition 2003 yielding a stop message because of configured virus detection rules, which it follows for the particular system that was going to be used to test Deep Freeze. However, once Norton Antivirus was disabled the install continued without any problems or errors. The corollary is: read and follow the instructions before beginning a new installation of any product.

After installation several tests were run using the program, including software installations, changing partitions, and deleting system files. The software Opera was installed while in frozen mode. When the system was restarted, it was as if the software was never installed. Also, the computer's hard drive was formatted in order to start everything fresh, and the program performed as expected. While in thawed mode, the computer was restarted and all changes made were saved correctly. Even after deleting all registry keys the deepfreeze was successful. All in all, the program performed as promised by the creator for all functions that it makes available.

Also there are definite advantages to installing the professional version rather than the standard version, especially if the program is being used on a network. The standard version does not offer the ability to have multiple configurations, such as each user on a network having their own Windows configuration files. This made it impossible to see how this is achieved. The standard version also does not allow the thawed space storage size to be altered. The maximum size is only 2 Gig and that is not enough space for some users.

### **Final Results/Recommendations**

The Deep Freeze utility basically allows computers to become testing machines that can discard changes made to systems without having to reformat hard drives or reinstall applications to get systems back to a known state. Deep Freeze adds security and standardization to the Windows OS platform. System default settings can not be permanently altered without proper authority. However, the utility excels by allowing users to add, delete, or modify systems at their own discretion, temporarily without damaging the system. In addition, when the system reboots, everything the user has done will be lost and the machine will be restored. System administrator's time is saved given the decrease in maintenance and assessment of individual workstations due to Deep Freezes' ability to restore computers back to a known state. The only downsides may be licensing costs and the fact that users may forget to disable or enable the utility which is a catch 22, but given the need for standardization, Deep Freeze is a worthwhile product.

Recommendations are as follows: Any computer that has the potential to be altered by mischief or ignorance should possibly be protected by Deep Freeze in any environment. In addition, publicly available computers, computer teaching labs, classrooms, libraries, and the like are examples of systems that should install this program. In a nutshell, Deep Freeze is potentially beneficial for all system owners/users because at some point a user will mistakenly alter files or

install a program that will break or severely decrease system functionality, forcing the need to restore the system to a known working state. Deep Freeze may be better suited for end users with systems at their homes and small businesses that lack IT departments that have strong backup and data recovery solutions. Conversely, Deep Freeze may not be a great asset in the corporate environment where systems are probably backed up daily and data recovery plans exists.

**References**

Deep Freeze - Complete Windows Protection. 2003. 6 June, 2003  
<<http://www.deepfreezeusa.com>>

**Project:** Firewall Scheduling Tool

**Author:** Mario Hankerson, Narsimha Baradi, Kao-Yee Chua

---

### **Problem Background**

Currently the Department of Computer Science has a computer lab in Wilson-Wallis room 6 for students to experiment with the installation, configuration, and maintenance of operating systems and software. Various servers and other software are installed and configured all the time. A major problem is that computers can be connected to the ETSU network. At times this situation has caused connectivity problems for other users in the building because of conflicting or misconfigured software installed in Wilson-Wallis 6.

### **Project Goals**

This project aims to create a system that allows an administrator to schedule the activation and deactivation of firewall rules. We could not find such an existing system, so we chose to implement our own. Because firewall administration can be daunting for beginning administrators, we built an interactive command-line interface to the scheduling tool.

### **Project Details**

We chose OpenBSD as it has a proven track record for secure default installs. The firewall software that we chose was Packet Filter, or pf, because of its existing integration into OpenBSD.

Because some ETSU network and Internet connectivity is desirable for the lab, we configured the firewall to allow only certain types of outgoing traffic. All other possibly troublesome network traffic would be filtered. Building and configuring the firewall was the first phase of our project. We found that creating rules for the firewall can be time-consuming and tedious. In addition, the current system does not have any way to schedule firewall rules. A scheduling system helps to reduce the amount of administration necessary to run the firewall.

### **Project Details**

The existing pf.conf allows administrators to specify what rules to allow and disallow on different network interfaces. Our system adds scheduling capabilities, and all other network traffic is blocked.

The system works by adding and removing rules. The user will add rules to the rules database via the interactive command line interface. Every day at an off-peak time the system checks the rules database to determine whether any rules need to be activated or deactivated. Once these are determined the program finds only the rules that should be active. It will generate an appropriate pf.conf file if any changes are necessary, and will then restart pf using the new rules. All changes are committed to the CVS archive weekly to provide an audit trail. Users should never have access to the rules database directly; all changes should be made through the interactive command line interface.

Rules contain the following information: the Internet protocol(s) to allow, the port number(s) to allow, the subnet(s) to allow, the date that the rule should be activated, the date that the rule

should be deactivated, the name of the administrator making the change, the name of the person the change was made for, and a flag. The flag indicates whether the rule is either pending activation or currently active.

The user begins the scheduling tool by running the `fwschedule` at the command prompt. The main menu displayed allows the user to begin configuring the firewall. The first option allows the user to add a new scheduled rule into the firewall. The second option allows the user to delete an existing scheduled rule. The third and fourth options allow the user to list either active or pending rules. Figure 1 shows an example screen shot of a new rule being added to the system.

```

Adding Rule # 2
Enter (I)ncoming or (O)utgoing for this rule [I, O, -1 to exit to main menu]:
I,O
Enter protocol type(s) [TCP, UDP, ICMP, -1 to exit to main menu]:
TCP,UDP
Enter port number(s) [1-65535 separated by commas, -1 to exit to main menu]:
53
Enter originating subnets separated by commas [x.x.x.x/mask, * for any, -1 to exit to main menu]:
*
Enter destination subnets [x.x.x.x/mask, * for any, -1 to exit to main menu]:
*
Enter date rule should be activated [MM/DD/YYYY, -1 to exit to main menu]:
05/01/2003
Enter date rule should be deactivated [MM/DD/YYYY, 0 for default of one week, -1 to exit to main menu]:
0
Enter your name (name of person adding this rule, -1 to exit to main menu):
Mario
Enter name of person this change was made for (-1 to exit to main menu):
Pfeiffer

Main Menu
1 - add a new rule
2 - delete an existing rule
3 - list active rules
4 - list pending rules
5 - exit
Main Menu Choice?

```

Figure 2.4. Screenshot showing user Mario allowing all DNS traffic for user Pfeiffer in dates ranging from 05/01/2003 to 05/08/2003.

A log file is generated every time changes are made to the rules database. This file is contained in `fwschedule.log`. Every week the log is committed to the CVS archive and reset. Administrators can check this file to track changes made to the database and to determine any problems with the scheduling of firewall rules.

**Lessons Learned/Problems**

We feel that our tool provides an easy way for administrators to configure firewall scheduling. For the future we recommend that the tool be tested more in order to determine its limits. We have not run the system for any extended amount of time, nor have we tested it to see how many rules it can handle effectively.

**Final Results/Recommendations**

We also have some ideas for expanding the tool. One idea is to make the firewall scheduling tool with a scriptable command-line interface in addition to the existing interactive interface to improve the potential efficiency of administration. Another idea is to add several presets into the administration tool so that users will not have to enter in TCP, UDP, and ICMP values manually. Such efforts would further reduce the amount of administration necessary for commonly used services that require complex firewall rules. If the lab will eventually contain servers or provide remote access then the tool will need to be expanded to add the ability to specify incoming traffic. Last we feel that we should write some official documentation such as quick start guides and tutorials so that administrators can easily use the tool.

**References**

- "Hardening OpenBSD Internet Servers Packet Filter and IP Filter on Non Firewalls". GeodSoft, LLC. 30 April 2003 <<http://geodsoft.com/howto/harden/OpenBSD/firewall.htm>>
- "Open BSD FAQ 6.2". [www.openbsd.org](http://www.openbsd.org). 4 April 2003. 30 April 2003  
<<http://www.openbsd.org/faq/faq6.html#6.2>>
- Coune, Wouter. "The OpenBSD Packet Filter HOWTO". 5 April 2002. 30 April 2003  
<<http://www.inebriated.demon.nl/pf-howto>>
- Tran, Hoang. "OpenBSD firewall using pf". 9 November 2002. 30 April 2003  
<<http://www.muine.org/~hoang/openpf.html>>

**Products:** BackOfficer Friendly, LaBrea Tarpit (for Windows), Honeyd, KFSensor

**Product Type:** Honeypot Software

**Author:** Steve Fritts

---

### **Problem Background**

The purpose of this project is to examine several different "honeypot" applications. I examined four honeypots: BackOfficer Friendly, LaBrea Tarpit (for Windows), Honeyd, and KFSensor. These four were selected primarily because they are all free. Honeyd, and LaBrea and BOF are open source applications, while KFSensor is a trial version of commercial software. I describe several aspects of each application, including installation and configuration, ease of use, key features, and potential problem. I included an overview of the testing steps I took to examine each application. Also included for each are sample screenshots or printouts of services provided.

### **Product Placement**

A "honeypot" is a machine or group of machines used to lure and entice attackers. Some honeypots are used for security purposes while others are used for research. Honeypots do not "do" anything to stop hackers from infiltrating a network, other than possibly occupy them for a period of time. They can, however, serve as listening tools which collect information for analysis, and can serve as warning systems.

Honeypots come in many "flavors". They can be standalone machines that offer a variety of services, sitting and waiting for attackers to infiltrate them. They can be a group of machines which work in tandem and allow researchers or network security personnel to observe hacker attack patterns, or they can be stand-alone software applications that emulate machines. This paper discusses honeypots of the latter type.

Honeypots can offer a wide variety of services. Typically, the more services are offered on a honeypot, the more useful it can be. However, added functionality often means added security risks, if hackers are able to bypass the traps of the honeypot and infiltrate the honeypot host's operating system. The honeypots discussed here are probably of the safer lot, because they do not actually allow any services (with the possible exception of Honeyd)--they merely emulate them.

### **Criteria**

#### Description of Testing Environment

I used two machines to complete this project. One machine has an AMD 2100+XP processor and 512 MB RAM running Windows XP Professional. The other machine is a Pentium II 400 MHz processor with 256MB RAM running Red Hat Linux 8.0. Both of these machines are set up on a small home network. Both machines use DHCP for network connectivity. The DHCP addresses are assigned by a Netgear RT311 router. The Windows box was used for installing and running BackOfficer Friendly, LaBrea Tarpit, and KFSensor, and was also used for network monitoring via WinDump. The Linux box was used for everything else.

## BackOfficer Friendly

NFR Security, Inc. ([www.nfr.com](http://www.nfr.com))

Product download: <http://www.nfr.com/products/bof/overview.shtml>

Product Overview: BackOfficer Friendly is called a "lightweight Windows HoneyPot" by its creator, NFR Security, Inc. Originally designed for detecting Back Orifice, the application has evolved so that it can now listen for attempted connections to several common ports, including HTTP, FTP and SMTP. BOF sits in the task bar system tray until it detects an incoming request on one of the ports it monitors, at which point it will pop-up on the user's screen, displaying a warning message as shown in Figure 1 below. In addition to displaying alerts, BOF can also send fake replies back to the sender of a request. The replies are different depending on the type of service requested. Telnet requests receive a "login" reply asking for a user ID and password. It replies to requests on port 80.

Lance Spitzner, author of "Honeypots: Tracking Hackers", says that BackOfficer Friendly is "a great place to start" for those new to honeypots. This product is certainly the smallest in functionality of any reviewed here.

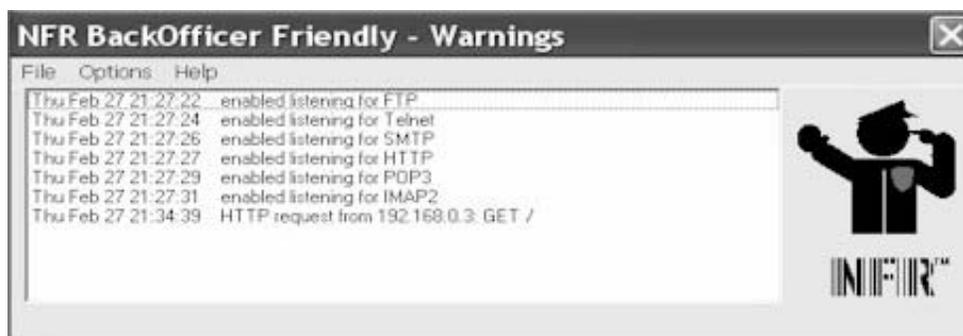


Figure 2.5: BackOfficer Friendly initial setup

Installation and configuration: Product installation was fairly easy. The self-installer was tarred, so I first had to download a utility that allowed me to untar the package in Windows. For this, I used WinRAR. After this task was completed, installing the software was simply a matter of double-clicking the self-installer.

Configuration of BackOfficer Friendly is a snap, mostly because there isn't much to configure. Figure 1 above shows almost all of the functionality available for this application. Clicking on "Options" in the menu bar allows me to check off which ports I want BackOfficer to listen to. I can also check whether or not I want BackOfficer to respond if it detects a call to one of the checked ports. Any traffic detected will appear in the window as shown in Figure 2.6 below.

Testing: To test the capabilities of BackOfficer Friendly, I first used nmap to run a port scan on the host machine to which BOF was installed. I scanned all 65,535 ports. Interestingly enough, BOF reported scans on all of the ports listed on NFR's website except for the one the product was designed for: Back Orifice (the default port for Back Orifice is 31337).



Figure 2.6: BackOfficer Friendly reporting port scans and telnet login requests.

I then tested the ability of BOF in receiving different types of connections, with and without 'Fake Replies' turned on. I received different results for HTTP requests ('401 Unauthorized' errors and 'the page cannot be displayed' errors respectively), telnet ('login' prompt and immediate 'connection to host lost' messages) and FTP ('503 Service Unavailable' errors and '421 service not available, remote server has closed connection' errors).

Figure 2 above shows several telnet connection attempts, which always appear the same in the BOF window--only the machine sending requests will get a different response depending on whether or not the 'Fake Replies' flag is turned on.

Other features: There is only one other feature available for BOF: it will save the results shown in the application window to a text file. The output for the text file is exactly what is shown in the BOF window (see Figure 2.6 above).

Ease of installation: *trivial*, however the application came tarred. I had to find a utility that would untar tar files in Windows before I could install it. For this, I used a free utility called WinRAR.

Ease of use. easy, again because of a lack of functionality here.

### **LaBrea Tarpit for Windows**

Created by Tom Liston ([labrea.sourceforge.net](http://labrea.sourceforge.net))

Product download: <http://www.hackbusters.net/LaBrea>

Product Overview: The "LaBrea Tarpit" honeypot was created by Tom Liston as a response to the "Code Red" worm. His original attempt, called "Code Redneck", was first created for UNIX systems and later became "LaBrea Tarpit". LaBrea Tarpit is now available in Windows as well as Linux versions.

LaBrea works by creating a subnet full of "virtual machines". It does this by examining the subnet of the host computer for existing physical machines attached to the subnet. It creates a virtual machine for any unused IP addresses on a particular subnet. It will then listen for connection attempts on any of the virtual machines (it will not respond to messages sent to physically present machines). If it receives messages on any of the virtual IP addresses, it will log the message as well as its reply. This is where the program gets interesting. Although LaBrea can be configured to respond to messages in several ways, it will typically reply with a SYN/ACK, which effectively ties up the machine sending the message.

Installation and configuration: Installation is a matter of downloading the file `LaBrea2_4Windows-exec-stable.exe` (for Windows). This is not a self-extracting file - this is the program executable! Configuring the program is not quite as straightforward, and will require the use of the man or help files at first. The program is run from the command line and can be configured via a (large) number of command line switches as well as via a file called `labrea.cfg`. In order to use LaBrea effectively, one must learn several of LaBrea's many switches. There are a ton of switches available for this program. Some of the more useful ones for a beginner like myself are included in Table 1 below:

**Table 1: Useful LaBrea Tarpit Switches**

<code>-o</code>	this switch sends output to stdout, which can then be piped to a text file
<code>-D</code>	this switch is useful for testing LaBrea's connectivity
<code>-v</code>	"verbose" - this instructs LaBrea to log more detailed activity information
<code>-z</code>	the <code>-z</code> switch must be used when running LaBrea on Windows machines
<code>-i</code>	the <code>-i</code> switch must be used. This specifies which Ethernet adapter LaBrea will run on. The syntax is <code>LaBrea -i [interface number]</code> .
<code>-p</code>	Using this switch causes LaBrea to permanently capture connections in a "persist" state, which, according to Liston, never times out. The syntax for this command is <code>LaBrea -p [maximum bandwidth rate]</code> . The maximum bandwidth rate specifies maximum bandwidth value you specify in bytes/second.
<code>-l</code>	send all messages to syslog

Testing: The first test I was able to successfully complete was using the `-D` switch, whose output is shown in Figure 2.7 below. This shows information about the host LaBrea is running on (including the host's MAC address) and can be used as a test to ensure LaBrea is able to connect to a network.

```
1. \Device\NPF_{BB5A72B6-6435-4EE6-9730-F0E70A398D3C} (Realtek 8139-series PCI NIC
(Microsoft's Packet Scheduler) )
```

Figure 2.7: Output of the command `LaBrea -D`. The `-D` switch shows information about the particular host you are running LaBrea on.

After verifying connectivity, I was able to run LaBrea. To start cranking out the virtual machines, I used the command `LaBrea -i 1 -z`. To test it, I then used `nmap` on a Linux box on my network to do a port scan of an IP address on my subnet which wasn't physically in use. The first `nmap` test was a port scan on phony IP address 192.168.0.6. `nmap` returned results showing all 65,535 ports on this IP address were open (see Figure 2.8 below). Figure 2.9 shows attempted connects to services on fake IP addresses 192.168.0.10 and 15. The result on my Linux box showed that connections to these fake IP address was established.

```
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 363
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 1518 *
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 1405
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 1489 *
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 34
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 329 *
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 229
Fri Feb 28 16:06:54 2003 Initial Connect (tarpitting): 192.168.0.3 36714 -> 192.168.0.6 6000 *
```

Figure 2.8: Results of an `nmap` port scan on fake host 192.168.0.6 as logged by LaBrea.

```
Wed Mar 05 06:31:01 2003 Initial Connect (tarpitting): 192.168.0.3 1043 -> 192.168.0.15 80
Wed Mar 05 06:31:01 2003 Additional Activity: 192.168.0.3 1043 -> 192.168.0.15 80 *
Wed Mar 05 06:31:04 2003 Additional Activity: 192.168.0.3 1043 -> 192.168.0.15 80
Wed Mar 05 06:31:07 2003 Additional Activity: 192.168.0.3 1043 -> 192.168.0.15 80 *
Wed Mar 05 06:31:13 2003 Additional Activity: 192.168.0.3 1043 -> 192.168.0.15 80
Wed Mar 05 06:31:54 2003 Initial Connect (tarpitting): 192.168.0.3 1045 -> 192.168.0.10 21 *
Wed Mar 05 06:31:54 2003 Additional Activity: 192.168.0.3 1045 -> 192.168.0.10 21
Wed Mar 05 06:31:57 2003 Capturing local IP: 192.168.0.15
Wed Mar 05 06:32:14 2003 Initial Connect (tarpitting): 192.168.0.3 1046 -> 192.168.0.10 23 *
Wed Mar 05 06:32:20 2003 Additional Activity: 192.168.0.3 1046 -> 192.168.0.10 23
Wed Mar 05 06:32:26 2003 Additional Activity: 192.168.0.3 1046 -> 192.168.0.10 23 *
Wed Mar 05 06:32:28 2003 Additional Activity: 192.168.0.3 1045 -> 192.168.0.10 21
Wed Mar 05 06:32:31 2003 Additional Activity: 192.168.0.3 1045 -> 192.168.0.10 21 *
Wed Mar 05 06:32:37 2003 Additional Activity: 192.168.0.3 1045 -> 192.168.0.10 21
Wed Mar 05 06:32:37 2003 Additional Activity: 192.168.0.3 1043 -> 192.168.0.15 80 *
```

Figure 2.9: Results of attempted connections to HTTP on fake host 192.168.0.15, as well as attempted telnet and FTP connections to fake host 192.168.0.10, as logged by LaBrea.

**Ease of installation:** *easy* - this was a snap for the Windows version - the app consists of one file to download, which is an executable.

**Ease of use.** *easy*, once you learn a few basic switches.

## Honeyd

Created by Niels Provos ([www.citi.umich.edu/u/provos/honeyd/](http://www.citi.umich.edu/u/provos/honeyd/))

Product download: <http://www.citi.umich.edu/u/provos/honeyd/>

Product Overview: honeyd is the most sophisticated honeypot tool I examined (and the coolest). It is similar to LaBrea in that it can create many virtual machines on unused IP addresses in your subnet. What is special about honeyd is that you can configure how the virtual machines "look" to probes and hackers.

Installation: honeyd is available in a pre-compiled version. If a person does not want to use the precompiled version, they could assemble and compile all of the components manually. Unfortunately, a key file was missing from the precompiled version (`xprobe2.conf`), so I had to figure out how to compile the program. Before compiling, you must make sure that these three items already exist on your Linux box: `libdnet`, `libevent`, and `libpcap` (these are networking libraries). Since I did not have these, I had to first download and compile all three of them. You also must make sure you have `arpd` installed on your Linux box. You must have `arpd` running before you can run honeyd.

Configuration: Configuring honeyd is not as easy as the other tools discussed in this paper. honeyd must run in sync with a daemon called 'arpd', which is used for APR spoofing. honeyd will act as though all ports on the subnet are open by default, but it can be configured to do a lot more. It can actually be configured to simulate being one or many of several types of devices: routers, network printers, Windows and UNIX hosts, even CRAY computers. But getting honeyd to correctly simulate these systems takes some planning as well as correctly updating the `honeyd.conf` file which tells honeyd how to respond when it receives a message aimed at one of these fake hosts. Not only that, but honeyd can actually call scripts which allow honeyd to respond differently to messages based on what is specified in the scripts. The templates can even be configured so that the virtual devices appear to "drop" packets, simulating real network activity.

```
### Windows computers
create windows
set windows personality "Windows NT 4.0 Server SP5-SP6"
set windows default tcp action reset
set windows default udp action reset
add windows tcp port 80 "perl scripts/iis-0.95/iisemul8.pl"
add windows tcp port 139 open
add windows tcp port 137 open
add windows udp port 137 open
add windows udp port 135 open
set windows uptime 3284460
bind 192.168.0.10 windows

### Linux 2.2.19 computer (Steve Fritts, 5 Mar 2003)
create linux
set linux personality "Linux 2.2.19 - 2.2.20"
set linux default tcp action reset
set linux default udp action reset
#set linux subsystem "/usr/sbin/httpd"
add linux tcp port 110 "sh scripts/pop3.sh"
add linux tcp port 25 "sh scripts/smtp.sh"
add linux tcp port 21 "sh scripts/ftp.sh"
set linux uptime 3284460
bind 192.168.0.11 linux

### Cisco router
create router
set router personality "Cisco IOS 11.3 - 12.0(11)"
set router default tcp action reset
set router default udp action reset
add router tcp port 23 "/usr/bin/perl scripts/router-telnet.pl"
set router uid 32767 gid 32767
set router uptime 1327650
bind 192.168.0.5 router
```

Figure 2.10: Here is a very small sample `honeyd.conf` file. This includes setups for a Windows 2000 IIS server and a Cisco router. The precompiled version of honeyd comes with literally hundreds of sample configurations.

Testing: Initially, I ran into problems with my honeyd setup (using the precompiled version). A file was missing from the honeyd tar package which instructs honeyd on how to respond to xprobe2 scans (the file is `xprobe2.conf`). I was unable to find the file, and was unable to compile the xprobe2 software on my Linux box. Compiling the honeyd program directly solved this problem. The following commands are used to start the process (note that arpd must be run first):

```
[root@dhcpc2 honeyd]# ./arpd
arpd[914]: listening on eth0: arp and not ether src 00:04:5a:43:ac:82
[root@dhcpc2 honeyd]# ./honeyd -l myLog.txt -f honeyd.conf
honeyd[916]: listening on eth0: ip and not ether src 00:04:5a:43:ac:82
```

The `-f` switch above directs honeyd to use predefined host signatures from the `honeyd.conf` file (as shown in Figure 6 above). The `-l` switch directs honeyd to log everything in the `myLog.txt` file. Once these daemons were running, I could then test connectivity. I did this first from my Windows XP box, as shown below (note that IP address 192.168.0.11 is bound to the Linux configuration shown in Figure 6):

```
C:\>ping 192.168.0.11
```

```
Pinging 192.168.0.11 with 32 bytes of data:
```

```
Request timed out.
Reply from 192.168.0.11: bytes=32 time<1ms TTL=255
Reply from 192.168.0.11: bytes=32 time<1ms TTL=255
Reply from 192.168.0.11: bytes=32 time<1ms TTL=255
```

```
Ping statistics for 192.168.0.11:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ftp 192.168.0.11
```

```
Connected to 192.168.0.11.
220 dhcpc2. FTP server (Version wu-2.6.0(5) Wed Mar  5 08:44:30 EST
2003) ready
.
User (192.168.0.11:(none)): admin
331 Password required for admin
Password:
530 Login incorrect.
Login failed.
ftp>
```

Ease of installation: *easy* for the precompiled version - it is simply a matter of unzipping, then untarring the software after downloading it to a machine.

Ease of configuration: *moderate*, based on what I described above. *difficult*, if you want your virtual machines to fool somebody who knows what they are doing. It is possible to design and create an entire virtual network with numerous servers, routers, printers and clients. You can be as elaborate with the configurations as you want. And, according to the honeyd MAN pages,

you can configure honeyd to run real applications on your Linux box using the virtual IP addresses you have set up.

### KFSensor

KeyFocus Ltd. ([www.keyfocus.net](http://www.keyfocus.net))

Product download: <http://www.keyfocus.net/kfsensor/>

**Product Overview:** KFSensor is another GUI-based honeypot application. It is limited in functionality, but is easy to install and configure. KFSensor really doesn't do anything that Honeyd or even LaBrea Tarpit does (other than the audible alarms). In fact, it is closer to BackOfficer Friendly in terms of functionality than the two command-line honeypots introduced in this paper. KFSensor is not an open-source product; I downloaded the "trial version" (however, I was not able to find a "commercial version" on KeyFocus' website).

**Installation:** Install of the trial version of KFSensor was simply a matter of downloading the trial-version executable and running it. Completion of the installation required a reboot of Windows, and when Windows came back up KFSensor was running.

**Configuration:** *easy* - KFSensor uses a simple GUI form that allows you to select a service, port number listening options, and banner responses. KFSensor is designed to only run on one machine--it emulates services, but is limited to emulating services only on the machine it is installed on. KFSensor does allow users to create "scenarios" where they can create mock services to listen on ports.

**Testing:** Testing began once again by using nmap from my Linux box. I started off with a port scan to all ports, which generated a long list of warnings in the KFSensor event window (Figure 2.11 below).

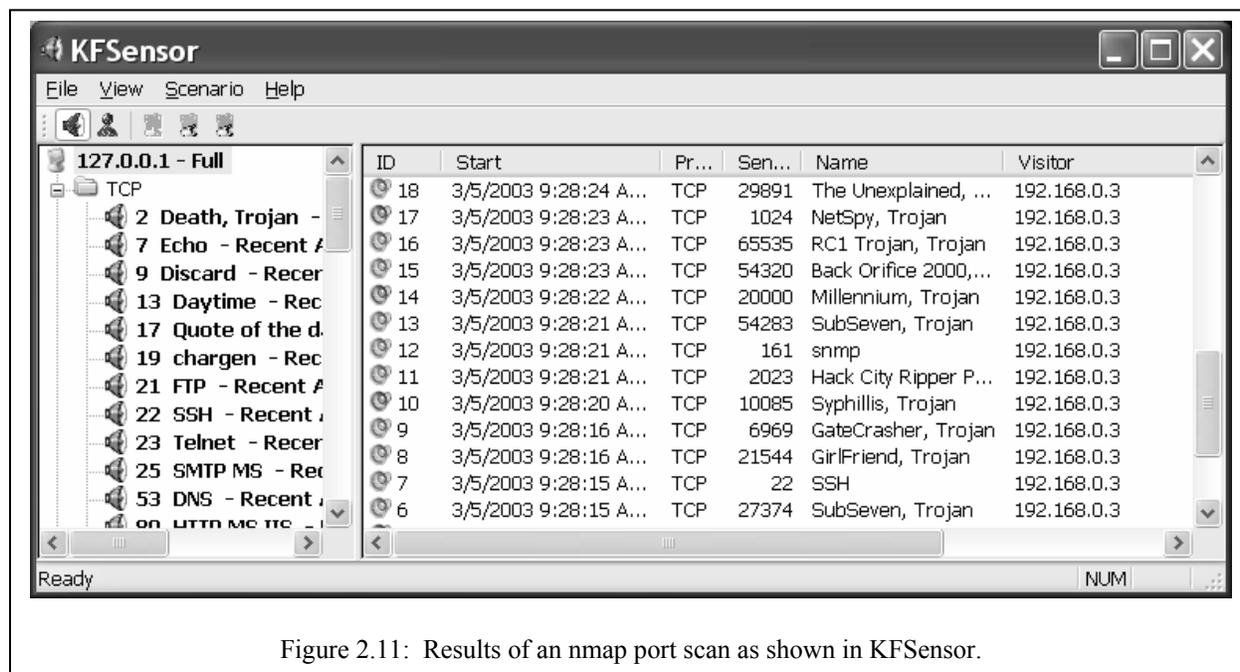


Figure 2.11: Results of an nmap port scan as shown in KFSensor.

After the port scan was completed, I tried some simple commands like *wget* and *telnet* to open services on the Windows box to see what kind of banners KFSensor would send back to me. KFSensor has only one response for each type of request. Figure 2.12 shows a response from an HTTP request using the *wget* command from my Linux box.

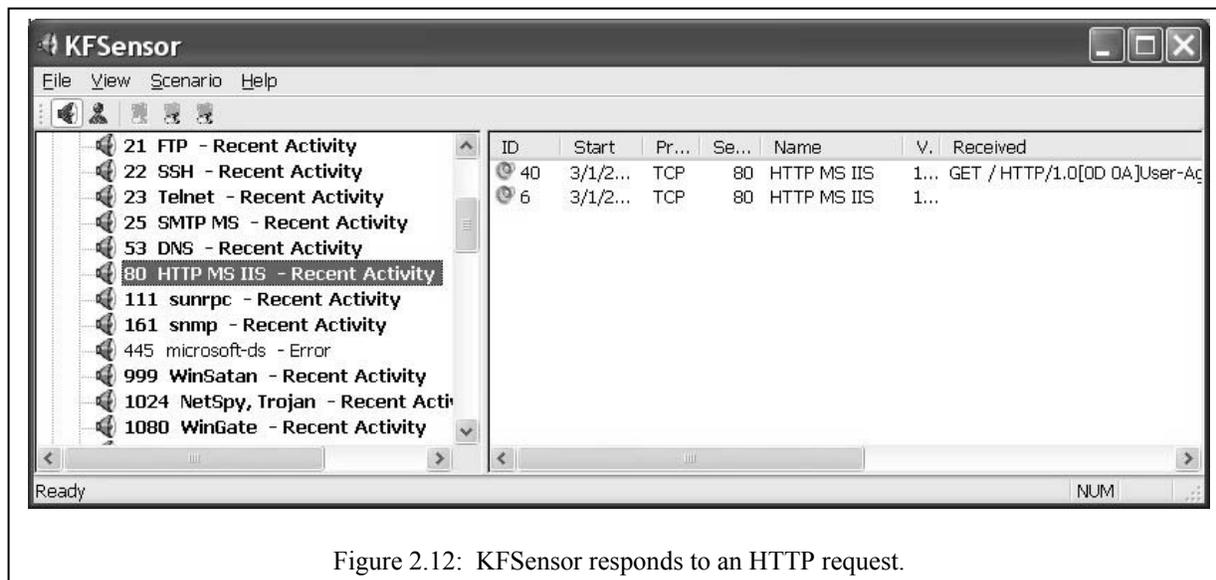


Figure 2.12: KFSensor responds to an HTTP request.

Ease of installation: *easy* - it is simply a matter of downloading the self-installing .exe file and rebooting Windows when the installation is finished.

Ease of configuration: *easy*, because of its simplicity. In a nutshell, KFSensor is a "beefed up" BackOfficer Friendly. Users can change the severity of alarms, save logs to a file, and set the ports that KFSensor listens to.

### Final Results/Recommendations

For individuals interested in learning about honeypots, these four are good ones to start with because of their simplicity. If I were going to learn to use all four, I would do so in this order:

1. BackOfficer Friendly
2. KFSensor
3. LaBrea Tarpit
4. Honeyd

BackOfficer Friendly is by far the easiest to use. KFSensor is next. These two programs run on Windows and use a GUI for user interaction. Both are limited in the services they provide. One major limitation of these applications is that they can only monitor incoming requests for the IP address of the machines they are installed on. BackOfficer friendly monitors incoming requests on a handful of ports, whereas KFSensor can monitor requests on a wide range of ports. KFSensor also allows users to configure the type of message that is returned.

Both KFSensor and BackOfficer Friendly allow users to save alerts to files. BackOfficer Friendly provides a visual alarm (popping up on the screen) when it receives messages, while KFSensor provides an audible alarm. Both of these applications can run in the Windows Taskbar System Tray and can be configured to start each time Windows reboots.

Even though the LaBrea Tarpit and Honeyd programs do not have a GUI, they far exceed BackOfficer Friendly and KFSensor in terms of functionality. The most important feature of these applications is that they can spoof any unused IP addresses on the subnet of the host they are installed on. Other features that these applications have in common are 1) the use of ARP to find unused IP addresses on the network, 2) the ability to send log messages to the syslog daemon, and 3) the ability to be added to crontab.

LaBrea's distinction among the four honeypots tested here is that it attempts to latch onto attacking hosts by keeping the TCP connections open (hence the name "tarpit"). These connections can be kept open indefinitely by using the *-p* (persist) flag.

Honeyd's distinction is that it can mimic a wide variety of operating systems and hardware signatures. In addition, Honeyd allows the user to integrate custom scripts that allow Honeyd's virtual hosts to reply to incoming requests in a variety of ways. Finally Honeyd allows some applications to run using the virtual IP addresses it has created.

**Table 2: Honeypot Functionality**

<b>Product</b>	<b>BackOfficer Friendly</b>	<b>LaBrea Tarpit (Windows)</b>	<b>Honeyd</b>	<b>KFSensor</b>
<b>Expandable</b>	no	yes	yes	yes
<b>Open Source</b>	no	yes	yes	no
<b>Logfile support</b>	yes	yes	yes	yes
<b>Notifications</b>	yes	no	yes	yes
<b>Ease of configuration</b>	trivial	easy	moderate - difficult	easy
<b>GUI</b>	yes	no	no	yes
<b>Command line</b>	no	yes	yes	no

This has been a fun and exciting project, and completing it has given me some insights into what honeypots are and how they work. I feel that the two GUI applications have little additional value, but I have only scratched the surface of what can be done with the two command-line applications, especially Honeyd. Because of their ability to capture all unused IP addresses on a subnet, I feel that both Honeyd and LaBrea can be valuable security tools, alerting system administrators even as they work to stall attackers.

In addition to learning about honeypots in general and how to use the four applications described in this paper, I have learned a lot about using Linux in general. Over the course of this project I have completed several "firsts" including: 1) using the *putty* application to remotely log into

Linux, 2) using *wget* and *ftp* to download files from the Internet using the Linux command line, 3) using *crontab* to schedule jobs, 4) compiling and building applications via the Linux command line, and 5) becoming much more familiar with *nmap*. In addition, I have learned more about the structure of directories and files in Red Hat Linux 8.0, and have learned to use *WinDump* to monitor network traffic (I used *WinDump* while testing *LaBrea* and *Honeyd*).

### Future Uses

I believe that there is potential to use the two command-line honeypots described in this paper in my thesis work on HIPAA security compliance. I plan to explore both *LaBrea* and *Honeyd* in much greater depth, and would like to plan a virtual network and use *Honeyd* to create it. I would like to configure the virtual hosts on the network to run actual services as described in the *Honeyd* documentation.

### References

- Baumann, Reto and Christian Plattner. "White Paper: Honeypots". 26 Feb. 2002.
- Liston, Tom. *LaBrea* "README.pdf" file. 4 Mar. 2003 <<http://www.hackbusters.net/LaBrea/README.pdf>>
- Liston, Tom. 'LaBrea - The Tarpit'. 15 Feb. 2003. 5 Mar. 2003
- NFR Security, Inc. 4 Mar. 2003 <<http://www.nfr.com/products/bof/overview.shtml>>
- Provos, Niels. *Honeyd* "README". 5 Mar. 2003
- Provos, Niels. "Honeyd: A Virtual Honeypot Daemon (Extended Abstract)". 2003
- Spitzner, Lance. "Open Source Honeypots: Learning with Honeyd". 20 Jan. 2003, 5 Mar. 2003 <<http://www.securityfocus.com/infocus/1659>>
- Spitzner, Lance. "Honeypots Solutions" 4 Mar. 2003 <<http://www.tracking-hackers.com/solutions/>>

**Product:** OpenAFS and Kerberos V5 integration

**Product Type:** Filesystem and Authentication

**Author:** Sai Divvala and Kao-Yee Chua

---

### **Problem Background**

OpenAFS, an open source implementation of AFS, has a built-in Kerberos server that implements the Kerberos Version 4 standard. However, the latest version of Kerberos is 5, and this version adds several new features and improvements including: new salts for more secure encryption, more flexibility in handling tickets and the ability to use custom encryption algorithms. We have an existing OpenAFS cell that currently uses the old Kerberos Version 4 standard. We would like to convert to use MIT's implementation of Kerberos V5 in order to take advantage of these new features.

Following is an overview of the differences between standard OpenAFS authentication, and OpenAFS authentication using Kerberos V5. First, a discussion of Kerberos authentication is necessary. A user initiates a session by running kinit on the client machine. The client then asks a key distribution center (KDC), a machine that runs Kerberos authentication, for a ticket. The KDC then generates a ticket granting ticket (TGT). This TGT is encrypted with the client's password, and sends the result to the client. The user enters his or her password, with which the client will then try to decrypt the encrypted TGT. If the decryption is successful, the client then keeps track of the TGT; this TGT is used as proof of the user's identity.

OpenAFS comes with a program called kaserver that implements the Kerberos V4 standard. This kaserver serves in the place of the KDC in a regular Kerberos system. AFS users log in using the aklog utility instead of kinit. Also, kaserver will use the AFS cell name as a salt to encrypt the TGT, which for OpenAFS is the appropriate token necessary to authenticate the user. If the client successfully decrypts the TGT, it then discards the result as it is only used for authentication. The client then takes the token and places it into the kernel so that the user can access files on his or her AFS cell.

In order to use Kerberos V5 a few changes must be made to the process. Users run the kinit utility into order to first log into the Kerberos realm. This part of the authentication process is done separately from AFS. After the user has received his or her TGT, he or she will then run aklog. This utility will examine the user's TGT, and send a request to the KDC for an AFS ticket based on these credentials. The server returns the ticket in Kerberos V5 format. Because AFS can only use the Kerberos V4 format aklog will send the ticket received to a Kerberos daemon. This daemon converts the ticket from V5 to V4 format, and then returns the ticket back to aklog on the client. The resulting ticket contains the tokens necessary for the user to access files on his or her AFS cell.

Integrating the two products would be ideal since OpenAFS uses the older version of Kerberos for authentication. To replace kaserver with Kerberos V5 would add allow better security and more functionality. In addition, once the migration is done, we can take advantage of improvements from future releases of Kerberos.

## Product Placement

OpenAFS and Kerberos are two products in wide deployment worldwide. The Andrew File System (AFS) was developed at Carnegie Mellon University. AFS is a distributed filesystem that promotes scalability, security, centralized management, transfer/migration capabilities, and location independence.

Kerberos, developed at MIT, is an authentication system that allows secure client/server communications across insecure connections. Sites that wish to use a single organization-wide strong authentication system often deploy Kerberos to help in the process of centralization.

## Installation Overview

In our first approach to integrating OpenAFS with Kerberos V5, we attempted several different methods to integrate Kerberos into an existing OpenAFS installation. Our efforts are documented in the Lessons Learned/Problems section.

Our second approach to integrating OpenAFS with Kerberos V5 worked successfully. A web page at [http://www.arayan.com/da/yazi/OpenAFS\\_Kerberos\\_5.html](http://www.arayan.com/da/yazi/OpenAFS_Kerberos_5.html) had step by step instructions on how to get the two to work on a vanilla Red Hat 8.0 system. First, we installed a clean copy of Red Hat 8.0 on a computer with server options. In addition we added the development packages and kernel development packages. Then we installed Kerberos, and last we installed OpenAFS. The integration itself happened in steps of the OpenAFS installation.

The Kerberos installation went by without many problems, as in our first approach we had installed and configured MIT's Kerberos distribution from scratch several times. Here we downloaded, rebuilt, and installed all packages from the Kerberos source RPM that were specifically created for RedHat 8.0. Then we first edited `/etc/krb5.conf` to include the name of our Kerberos realm, WWETSU.EDU (the Kerberos realm must always be in capital letters to work properly). We then did the same for `/var/kerberos/krb5kdc/kdc.conf`, and then added `des3-hmac-sha1` as the `master_key_type`. Then we added `des3-hmac-sha1:normal` to `supported_etypes`. Next was the creation of the Kerberos database using `kdb5_util`. We edited `/var/kerberos/krb5kdc/kadm5.acl` to reflect our realms. Then we added regular and administrative users for the realm: `sai`, `kao`, `admin`, `sai/admin`, `kao/admin`, `admin/admin`. Next the `afs` users (`afs` and `afs/admin`) were created using the `des-cbc-crc:v4` encryption type for AFS compatibility. The last step during the Kerberos installation was to start up the `kadmin`, `krb5kdc`, and `krb524` daemons.

The last step in the integration was the installation of OpenAFS. We downloaded, built and installed all packages from the OpenAFS source RPM using the 8.0.1 version obtained from the OpenAFS web site. We also had to install the `e2fsprogs-devel` package, the dependency we had for the OpenAFS RPM. Next we added our cell name, `wwetsu.edu`, to `/usr/vice/etc/CellServDB` and `/usr/vice/etc/ThisCell`. Next we turned off `AFS_CLIENT` and `AFS_SERVER` while performing the initial configurations. Third step was to start the `afs` daemons. Fourth, we ran the `bossserver` using the `-noauth` parameter so that we could perform initial configurations. We set the cell name, and created the server instances. Here we ran into problems with the domain name resolution. Since we were not running DNS, we added the appropriate entries to the `/etc/host`. In production systems the system would be

pointed to an appropriate DNS server. The next step was important for the integration to work. Here we ran `kadmin.local` and added the `afs` user to the keytab entry so that AFS could authenticate using Kerberos V5. We then imported the key into AFS using `asetkey`. We created the admin user and created fileserver instances. When the tutorial had us create a `root.afs`, we ran into a problem. The documentation did not have us create a partition specifically for `/vicepaa`, and so we had to implement a workaround. We moved the home partition (`/dev/hda3`) to the root partition (`/dev/hda2`). Then we formatted `/dev/hda3` and mounted it onto `/vicepaa`. The next step was to restart everything, so we killed `bosserv`, turned on `AFS_CLIENT` and `AFS_SERVER` in `/etc/sysconfig/afs`, stopped AFS, and removed the `libafs-2.4.18-19.8.0-i686` module. We restarted by running `/etc/rc.d/init.d/afs start`. Then we logged into the Kerberos realm and ran `aklog` as admin to set some permissions and work with the filesystem. Replication points were added. The last step was to add the groups and users. We then created a user along with the corresponding home directory. We logged into the Kerberos realm, ran `aklog`, and typed tokens. Everything worked, and we tested the cell by creating a file, logging out, doing `kinit` and `aklog` again. The file was there, and we wrapped up the project.

### Lessons Learned/Problems

We never successfully migrated the existing OpenAFS cell to use Kerberos V5 authentication in our first approach to the problem. Our goal was to get a working version of the `aklog` utility. `Aklog` would read a Kerberos TGT and get the appropriate AFS tokens for the authenticated user. We tried several different approaches to this: using the AFS to Kerberos migration kit by Ken Hornstein, using source code, and using precompiled binaries. Each of these steps proved to lead to either no progress or only partial success.

Ken Hornstein's Migration Kit is well known, and it contains a lot of extremely useful information on how AFS performs authentication. The version 1.3 kit comes with patches, documentation and source code that help AFS system administrators to use their existing AFS cells with Kerberos V5 implementations. We used the documentation to successfully configure Kerberos and OpenAFS with the appropriate accounts and tickets necessary for successful migration. Unfortunately the patch supplied was only applicable to version 1.0.6 or Kerberos, and we tried to use the latest version, 1.2.7. We attempted to compile the migration utilities from the source code, but ran into a lot of problems with missing include files. We thought the problem might be that the include files necessary to compile AFS programs were missing. Here we tried to examine the contents of the AFS RPM, but could not find a way to install only the include files. The documentation was extremely helpful in understanding the theory behind the migration, but we never compiled `aklog` successfully.

Second we tried to use a source distribution from MIT. This package was located at <http://web.mit.edu/openafs/>. Although the source code was different, we ran into compilation problems that we encountered with the AFS Migration Kit.

Last on the agenda was using precompiled binaries. A person at UIUC maintains binaries for various operating systems and architectures for Hornstein's AFS Migration Kit at <http://ismene.csl.uiuc.edu/afs-mig/>. We downloaded the Linux binaries and tried to use the `aklog` provided. However we could never get the file to get AFS tickets. The program gave an error

indicating that there were problems with an unsupported encryption type. We tried to add some more encryption types, but none worked with the binary. Unfortunately we could not make changes to the program, and so our last attempt during the migration to get aklog to work was unsuccessful.

After all of our efforts, we learned a few lessons. First of all, for successful installation of software packages version management is of the utmost importance. Mixing different versions of software and libraries always seemed to cause problems for us.. Another lesson hard learned was that Linux RPM installations can be extremely difficult to manage because of problems with package dependencies. Sometimes we found that we had to download and install three or more packages just to the desired one working. Fortunately the rpmfind service exists to allow access to RPMs. Last, we found that even precompiled binaries don't always work. Different environments will yield different results for different systems.

### **Final Results/Recommendations**

Fortunately we finally got the OpenAFS cell to use Kerberos V5 for authentication. Both systems are complex, and therefore integration of the two is a huge task. We suspect that our first approach did not work because while Kerberos was built from scratch, OpenAFS was installed using precompiled binaries. This probably introduced some library and encryption incompatibilities. In the future we highly recommend installing both of these from scratch for the target system. Using RPMs that are made for the same system is most likely the best method to pursue since the probability of incompatibilities will be higher. However, if this is not possible, then by compiling from source the appropriate dependencies and incompatibilities can be worked out at build time. Overall this has been a very educational and challenging project.

### **References**

- "AFS-Kerberos 5 Migration Kit". 30 April 2003 <<http://ismene.csl.uiuc.edu/afs-mig>>
- "Kerberos: The Network Authentication Protocol". 9 April 2003. 30 April 2003  
<<http://web.mit.edu/kerberos/www>>
- "OpenAFS, Kerberos 5, LDAP and Linux". 12 February 2003. 30 April 2003  
<[http://www.arayan.com/da/yazi/OpenAFS\\_Kerberos\\_5.html](http://www.arayan.com/da/yazi/OpenAFS_Kerberos_5.html)>
- Frequently Asked Questions About Kerberos "Kerberos FAQ". 30 April 2003  
<<http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html>>
- OpenAFS. 18 April 2003. 30 April 2003 <<http://www.openafs.org>>

**Product:** Wilson Wallis OpenBSD Firewall Project

**Product Type:** Operating System/Firewall Installation

**Author:** Narsimha Baradi, Mario Hankerson, Kao-Yee Chua

---

### **Problem Background**

The computer lab at Wilson Wallis is frequently used by several faculty and students. Faculty use computing resources in the lab as part of instruction for numerous classes through each school day. The tasks are many; some examples are installing operating systems, installing applications, building computers and configuring network equipment. In addition the lab can be connected to the ETSU network, and therefore can connect to the Internet. Such an environment can be highly prone to malicious behavior. Malicious users might use lab computers to hack into other systems outside of the lab or run unauthorized servers. These can result in a waste of the university's network resources, and may open up potential legal problems. For this reason we decided that a firewall should be installed into the lab. The firewall will allow the administrators of the lab to regulate who will have access to the Internet. Access control will be provided on a port by port basis. In addition administrators will have the ability to track any unauthorized outgoing network traffic. Our project is divided into two parts. The first involves setting up the firewall. The second part involves automating the firewall's operation, and setting up log analysis facilities.

### **Product Placement**

According to its homepage, OpenBSD has had “only one remote hole in the default install, in more than 7 years!” This reputation is echoed throughout industry. The developers are committed to security. The Packet Filter (pf) firewall program is tightly integrated into OpenBSD, and therefore is a natural choice for this installation.

### **Installation Overview**

The first part of the project involves setting up the firewall, and is mostly finished. The first task was to gather requirements from both the instructor and the Office of Information Technology (OIT). The instructor specified the OpenBSD platform for the firewall because of its track record for good security. It was also a good choice because it comes by default with firewalling capabilities with its packet filter (pf) program. Speaking with the computer lab manager at OIT only yielded a link to ETSU's Code of Ethics at <http://www.etsu.edu/humanres/ppp/PPP-44.htm>. The networking contact did not specify any requirements except that we would use DHCP to connect to the campus network and that our firewall would not affect any other clients.

In order to do the installation we began by burning a CD with the base packages for the i386 platform. Then we created a boot floppy for our system. The installation happened quickly and smoothly on our system with two network cards. Default OpenBSD installs only have the ssh service installed, and so the first step was to disable remote root ssh logins. Then we added administrative users to the system. In order to allow them to login as root using the su command, we added the users to the wheel group. We then hardened the services offered by default by following the guide at <http://geodsoft.com/howto/harden/OpenBSD/services.htm>. The system did not install the network time protocol (ntp) by default, so we installed it from the source code. In addition, we downloaded the ntpd source to the kernel and installed a stripped down version.

Some changes were necessary to use the system as a firewall. Also, we configured the pf firewall configuration file (`/etc/pf.conf`) with the rules for our system. One of the network interfaces will connect to the ETSU network, and thus is configured to use DHCP for network configuration. The other will connect to the main switches of the Wilson Wallis lab.

The second part of the project involves automating various tasks associated with running the firewall. As mentioned earlier, the system will be written in perl. Perl was chosen because the project will require a solution that is more heavyweight than shell scripting, but lighter weight than a full blown compiled language. Following is our proposed design of the logging and automation facilities of the firewall.

### **Lessons Learned/Problems**

The installation of OpenBSD and pf was straightforward. The most difficult part of the process was finding hardware that OpenBSD supports. Even after a few efforts we could not get two different CD-ROM drives to work in order to install the system. Fortunately the important hardware such as the network cards and video card were supported. In the future we highly recommend that interested users check the hardware compatibility list before installing OpenBSD.

### **Final Results/Recommendations**

Packet filter has the ability to log all packets that the firewall drops. These logs will be kept to audit both incoming and outgoing traffic that is denied. We are looking at existing solutions that analyze pf logs. The two solutions that we came up with are fwanalog and pfstat. The results of the analysis will be e-mailed to the administrators on a regular basis.

The automation of the firewall will be accomplished by creating a system that keeps track of how long ports should be open. By default no outgoing and incoming ports will be allowed except incoming ssh access for administrators. Administrators will have the ability to specify a rule. Rules include the particular port numbers and type (UDP, TCP or ICMP) to allow. Also, each rule will have a time (in days) that the rule will apply. If an administrator creates a rule and no time is specified then by default the rule will apply for one week.

An original firewall configuration allowing no ports except ssh will be created. Version control will be done via a CVS repository installed on the machine. All changes to the configuration file will be committed to the repository. Doing so will provide a means to audit changes in case any malicious changes are made to the firewall. If any rules are changed, a new configuration file will be generated and committed. A series of daily cron jobs will check to determine whether to turn a generated rule on or off.

An example user scenario follows. A firewall administrator receives a request to open TCP port 7000 for 10 days. The first step is to ssh into the firewall using his or her user account. Then the administrators will su to root, and then run the scripts that create rules. The administrator will then create a new rule to allow TCP port 7000 to be open for 10 days. The system will then generate a new pf.conf file, and commit both the old and new versions of the file to the CVS repository. Pf will then be run using the new configuration.

**References**

- "Hardening OpenBSD Internet Servers Packet Filter and IP Filter on Non Firewalls". GeodSoft, LLC. 30 April 2003 <<http://geodsoft.com/howto/harden/OpenBSD/firewall.htm>>
- "Open BSD FAQ 6.2". [www.openbsd.org](http://www.openbsd.org). 4 April 2003. 30 April 2003 <<http://www.openbsd.org/faq/faq6.html#6.2>>
- Tran, Hoang. "OpenBSD firewall using pf". 9 November 2002. 30 April 2003 <<http://www.muine.org/~hoang/openpf.html>>

**Product:** Security-Enhanced Linux  
**Product Type:** Operating System  
**Author:** Trey Buck

---

### **Problem Background**

Over the past several decades, the face of computer system security has changed dramatically. Early operating systems had no security whatsoever, only those individuals who had a key to the computer room were able to access the system. System security today is a vast improvement over earlier systems. Most modern operating systems now enforce mandatory authorization and authentication, and limit the rights of users to those rights they need to perform their assigned tasks.

Yet, despite the improvement in system security over the years, security is still a major issue. Systems are still vulnerable to a variety of pitfalls. Runaway processes, break-ins, and Denial of Service attacks still plague most operating systems because the security measures provided cannot effectively deal with these situations.

The National Security Agency (NSA) recognized the insecure nature of modern operating systems and began a project to demonstrate how system security could be improved. The NSA, along with three other organizations<sup>3</sup>, created a prototype operating system with security measures based on a new paradigm.

### **Product Placement**

Security-Enhanced Linux (SELinux) is a prototype operating system based on the Linux operating system kernel. The original kernel source code was modified to include mandatory access control. Mandatory access control is a critical security feature needed to separate information based on security requirements. The main goal of system security is exactly that: provide system entities with the minimal amount of information and resources required to complete the task.

The architecture provided by SELinux dictates that security restrictions be enforced by policy, not the attributes of system objects. Security settings are centralized by a selection of configuration files rather than object specific attributes, such as file permissions. In this way, the operating system has a clean separation between policy and enforcement. Separation between policy and enforcement results in a security architecture that is more effective and easier to administrate. This architecture provides fine-grain control over process initialization and inheritance, program execution, file systems, directories, files, open file descriptions, sockets, messages, and network interfaces.

SELinux is a prototype. It is a proof-of-concept that is intended to demonstrate how mandatory access controls can be implemented in a Linux environment. However, it is a fully-functional system that maintains binary compatibility with existing applications, and source compatibility

---

<sup>3</sup> The original contributors were the NSA, Network Associates Laboratories, The MITRE Corporation, and the Secure Computing Corporation.

with existing kernel modules. The SELinux distribution is essentially a security patched kernel and a number of security patched tools and utilities.

### Installation Overview

SELinux is designed to be installed on an existing Linux system. It is designed to be installed on RedHat 7.2 or 7.3. Here, an attempt is made to install SELinux on RedHat 8.0. The first step is to download the SELinux distribution from NSA's web site, <http://www.nsa.gov/selinux>. The installation tree is provided in a compressed file and is the easiest way to obtain the needed files. After decompression, the first step is to ensure the various configuration files are consistent with your machine's file structure. These configuration files expect to find a certain directory structure and must be modified if the paths are inconsistent. Installation on RedHat 8.0 did not require changes to these files.

SELinux does not play nicely with most display managers. Display managers must not be configured to start at boot time. For RedHat, this means setting the initial run level to 3 in `/etc/inittab`, and commenting out the `[gk]dm` entry in `/etc/rc.local`. Also, the RedHat default development packages should be installed on the system as the binaries must be compiled from source and depend on the `gcc` and `ncurses` packages. This is not obvious after a review of the included documentation.

Ideally, SELinux can be installed in its entirety by running the command `make quickinstall` from the SELinux directory of the installation tree. Unfortunately, this is a very lengthy and processor intensive task and may fail if the system begins to run out of resources. Should this occur, the system must be built and installed manually. Building and installing manually was required for this installation. The exact commands are not listed here as they are well documented in the readme file. Below are the steps taken to install SELinux, problems that were encountered, and how they were overcome.

1. The kernel was first built by issuing the appropriate commands. A kernel configuration menu is presented during this operation. Here, the kernel compilation options were specified. In particular, the processor type, network device, and `ext3` options were set to the appropriate values.
2. The installation was begun by first attempting to install the kernel. The first try was unsuccessful as the script was trying to locate the nonexistent `/lib/modules/2.4.20-lsm1` directory. The directory in question was misnamed earlier in the script to `/lib/modules/2.4.20-selinux`. Renaming the directory solved the problem.
3. The support files were installed.
4. The policy files for the optional tools were installed. This step may be omitted if not installing the tools.
5. The example policy file was built and installed. There was a problem with this portion of the installation. The documentation indicated the command `make install-src install` should be executed. The makefile's `install-src` target expects the directory `/etc/security/selinux/svc/policy` to exist. This directory, however, is created by

the makefile's install target. The arguments to make should be reversed and the command should be issued as `make install install-src`.

6. libsecure and its test programs were successfully built and installed.
7. The modified applications were installed with effort. This is a rather lengthy process and the build failed once due to lack of system resources. A second attempt was successful.
8. Both the optional tools and Selopt utilities were not installed due to the problems encountered previously.
9. The example files were copied from the `utils/appconfig` directory to the `/etc/security` directory.
10. The setfiles component was successfully built and installed.
11. The persistent label mappings were successfully initialized
12. The bootloader was configured to use the SELinux kernel by issuing the command `/sbin/grubby --add-kernel=/boot/vmlinuz-2.4.20-selinux --copy-default --make-default --title "SELinux"`. This is the appropriate command when using GRUB as the bootloader. LILO setup is different and described in the readme.
13. The system was rebooted.

At this point, SELinux is installed on the system. Booting into the new kernel however, presented a bit of a challenge. After rebooting the system the bootloader was unable to mount the root filesystem. The system was reset and the original kernel booted. Since the root filesystem was an ext3fs filesystem, the kernel compilation options were reexamined to ensure ext3fs support was included, which it was. An extensive search of various mailing lists revealed a possible solution: explicitly define in `/boot/grub/grub.conf` on which disk partition the bootloader should look for the root filesystem. The SELinux entry in the `grub.conf` file was modified. The `root=LABEL=/` parameter was replaced with `root=/dev/hda2`.

Since the system had been booted into a non-SELinux kernel, the persistent label mappings were reinitialized by running setfiles again. The system was then reset and booting to the SELinux kernel was successful.

The access policies are determined by a number of configuration files located in the `/etc/security/selinux` directory. There are configuration files for users, domains, file contexts, types, etc. These are the abstract objects used by the SELinux architecture. To make a change in policy, the appropriate configuration file needs to be edited and the policy rebuilt, reinstalled, and reloaded. A make file is provided that will build, install, and load the new security policy: `make load`.

Having the policy files in one place is excellent for configuration management. Customizing the files, however, can be tricky in some cases. Some of the configuration files are quite complex and it is not always obvious which file or portion thereof needs to be edited to achieve the desired behavior. The most important files are `users` and `file_contexts/types.fc`. The

users file lists the users on the system and their authorized roles. Typical entries are similar to the following:

```
user root roles { user_r sysadm_r }
```

This entry indicates the root user can take on the user\_r role (an underprivileged user) or the sysadm\_r (an administrative role). The file\_contexts/types.fc file indicates the context of filesystem objects. This includes files, folders, and devices. The context of the object determines to a large degree who is allowed to access and use it. An entry typically consists of a regular expression and the associated contexts. For example, the following are the entries that indicate the context for objects in the /root directory, including the /root directory itself.

```
/root/(/*.*)?          system_u:object_r:sysadm_home_t
/root/.ssh(/*.*)?      system_u:object_r:sysadm_home_ssh_t
/root                  system_u:object_r:sysadm_home_dir_t
```

Some applications have file dispersed throughout the filesystem which must run under specific contexts not specified in file\_contexts/types.fc. For these situations, specific contexts for an application's files can be configured in a separate file. These files are located in the file\_contexts/programs directory. The contents of traceroute.fc are listed here as a simple example

```
#traceroute
/usr/bin/traceroute    system_u:object_r:traceroute_exec_t
/usr/sbin/traceroute   system_u:object_r:traceroute_exec_t
```

If the policy files are not configured correctly, it is possible the system will be unusable. This does not however mean the system cannot be recovered. The security mechanisms in SELinux are dependent upon the SELinux kernel. Booting to a different kernel will allow an administrator to reconfigure or restore the policy files in order to recover the system. During testing, it is recommended that the original kernel be kept on hand for just such a case. Once the system is recovered, the persistent label mappings must be reset before booting into the SELinux kernel.

### Lessons Learned/Problems

The major problems encountered are described above. None of the setbacks were insurmountable, but debugging and finding solutions to them resulted in a significantly more complicated the installation procedure. It was discovered later that patches for the installation scripts had been distributed that solve some of the above problems. It is unclear however, why the system killed the build process at several points. The RedHat system was freshly installed, so the chances of a rouge program consuming resources is slight. The symptoms are indicative of memory not being freed at some point during the installation.

As described above, the entire SELinux security architecture can be circumvented by booting to a non-SELinux kernel. For this reason, physical security is very important if the integrity of the security architecture is to be maintained.

**Final Results/Recommendations**

The installation of SELinux is not recommended for those with little Linux experience. If the phrase 'kernel configuration options' is not familiar to you, think twice before attempting this install. Installing the system can be tricky and there is no limit to the problems that may be encountered. In other words, installing SELinux on a critical server is not a good idea.

As mentioned before, SELinux is a proof-of-concept and should not be relied upon in a mission-critical environment. Nevertheless, it is well worth examining as it provides a different view of system security and is a good start for building more secure operating systems in the future.

**References**

- "Security-Enhanced Linux". National Security Agency. 29 April 2003 <<http://www.nsa.gov/selinux>>
- Coker, Russell 29 April 2003 <<http://www.coker.com.au/selinux/>>
- Sourceforge.net 29 April 2003 <<http://selinux.sourceforge.net>>

**Product:** Tripwire for Linux 8.0

**Product Type:** Utility

**Author:** David Frazier

---

### **Problem Background**

Tripwire is product designed to let a system administrator know if an intrusion has occurred on any of their machines. It does this by creating a database of checksums of files on the system. This database is encrypted to make sure that hackers are not able to modify it. One common trick that hackers use to escape notice is to install Trojan versions of common commands such as ls or more. Tripwire lets the SA know this has happened by checking all files against the database to see if any have changed.

Tripwire includes a policy definition language that allows the user to customize which files need to be watched. For example, some files will be expected to change frequently. The Policy configuration file can be written to have Tripwire ignore these files. The Policy language can also be used to create a hierarchy of violations. For the most serious of violations, Tripwire can email the SA.

### **Product Placement**

Tripwire exists for most versions of Unix, and well as for Windows NT. Tripwire is a commercial product on those platforms. Tripwire is also available in an Open Source version for Linux. Tripwire is now included with most versions of Linux. It is not installed by default, but is on one of the installation cds.

### **Installation Overview**

I installed Tripwire on a computer running Redhat 8.0. Tripwire was included as a RPM package on one of the distribution cds. It was trivial to install the program using RPM. The program is placed in `/etc/tripwire`. Root access is needed to install or use Tripwire for obvious security reasons.

After the program is installed, you must create the database. This is easy to do with the `tripwire --init` command. This command asks you for a host key and a site key. These are passwords entered by the user that are used to sign the database. This encryption scheme makes it difficult to change the database without the site key. You are asked for the site key whenever you are running Tripwire. The database takes a snapshot of the files determined to be critical based on the policy file.

An integrity check is automatically run once a day using a cron job. It is also possible to do an integrity check whenever you want by issuing the `tripwire -- check` command. When you run this command, violations are printed to the screen, and an encrypted copy of the report is saved for future viewing in the `/var/lib/tripwire/report` directory.

The policy information is kept in a text file called `twpol.txt`. It is in this file that you can customize what Tripwire considers a violation. One common change that needs to be made to this file is to comment out the lines where Tripwire looks for files that your system does not have

installed. These show up `File Not Found` errors in the Tripwire reports. This should be corrected, so that in the case of a needed file being deleted, it will not get hidden in a list of file that were never there. You also need to add any files that should be there, but that are not listed in the default policy file. An abbreviated sample `twpol.txt` is listed in Appendix 1.

Files are divided into categories. The following classifications are used:

```
SEC_CRIT          # Critical files that cannot change
SEC_SUID          # Binaries with the SUID or SGID flags set
SEC_BIN          # Binaries that should not change
SEC_CONFIG       # Config files that are changed infrequently but accessed often
SEC_LOG          # Files that grow, but that should never change ownership
SEC_INVARIANT    # Directories that should never change permission or ownership
SIG_LOW         # Non-critical files that are of minimal security impact
SIG_MED         # Non-critical files that are of significant security impact
SIG_HI          # Critical files that are significant points of vulnerability
```

Rules are set using the following syntax:

```
(
  rulename = "Descriptive name of rule to be shown in report"
  severity = $(SIG_HI, SIG_MED or SIG_LOW) #Determines the severity
of the violation.
)
```

Files listed along with what category they are in. For example:

```
/sbin/accton          -> $(SEC_CRIT) ;
/sbin/badblocks       -> $(SEC_CRIT) ;
/sbin/busybox         -> $(SEC_CRIT) ;
```

After the severity is listed, you can optionally add an `mailto` parameter to cause Tripwire to email someone if this violation occurs.

The `twpol.txt` is only a copy of the policy in force; it is not the policy itself. To update the policy that Tripwire uses, you must first change the `twpol.txt` file, and then run the following command to encrypt the `twpol.txt` file and put it in force:

```
Twadadmin -create-polfile -S site.key /etc/tripwire/twpol.txt
```

Note that the `site.key` is required in this command. This insures that only an authorized person can change the policy file. Once the policy is correct, a report should be generated daily, and any flagged violations will be emailed to the SA.

The reports can be viewed using the `twprint` command. These reports are encrypted, and thus you must enter the site key to be able to view them. A sample report from my system is listed in Appendix 2.

**Lessons Learned/Problems**

It is a good idea to do an integrity check when you install the system, before making policy changes, just to get a feel for what is being checked for. If violations are found on a clean system, before anyone else can access the system, you have false positives. You will need to adjust your policies. The report should contain no violations on its initial run on a clean system.

**Final Results/Recommendations**

One of the keys to using Tripwire effectively is to create the database when the system has just been installed, and before it is on the network. As soon as a system is connected to the Internet, there is the possibility that files will be altered. Creating the database is relatively easy. Using Tripwire is often mentioned in lists of best security practices.

## Appendix 1 – Abbreviated Policy File

```
#####
#
#           Policy file for Red Hat Linux
#                   V1.2.0rh
#                   August 9, 2001
#####
#####
# Global Variable Definitions
# These are defined at install time by the installation script.  You may
# Manually edit these if you are using this file directly and not from the
# installation script itself.
#####
@@section GLOBAL
TWROOT=/usr/sbin;
TWBIN=/usr/sbin;
TWPOL="/etc/tripwire";
TWDB="/var/lib/tripwire";
TWSKEY="/etc/tripwire";
TWLKEY="/etc/tripwire";
TWREPORT="/var/lib/tripwire/report";
HOSTNAME=localhost;
@@section FS
SEC_CRIT      = $(IgnoreNone)-SHa ; # Critical files that cannot change
SEC_SUID      = $(IgnoreNone)-SHa ; # Binaries with the SUID or SGID flags set
SEC_BIN       = $(ReadOnly) ;      # Binaries that should not change
SEC_CONFIG    = $(Dynamic) ;       # Config files that are changed infrequently but
accessed often
SEC_LOG       = $(Growing) ;       # Files that grow, but that should never change
ownership
SEC_INVARIANT = +tpug ;            # Directories that should never change permission
or ownership
SIG_LOW       = 33 ;               # Non-critical files that are of minimal security
impact
SIG_MED       = 66 ;               # Non-critical files that are of significant
security impact
SIG_HI        = 100 ;              # Critical files that are significant points of
vulnerability

# Tripwire Binaries
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI)
)
{
  $(TWBIN)/siggen          -> $(SEC_BIN) ;
  $(TWBIN)/tripwire       -> $(SEC_BIN) ;
  $(TWBIN)/twadmin        -> $(SEC_BIN) ;
  $(TWBIN)/twprint        -> $(SEC_BIN) ;
}

# Tripwire Data Files - Configuration Files, Policy Files, Keys,
Reports, Databases
(
  rulename = "Tripwire Data Files",
  severity = $(SIG_HI)
)
{
  $(TWDB)                  -> $(SEC_CONFIG) -i ;
  $(TWPOL)/tw.pol          -> $(SEC_BIN) -i ;
  $(TWPOL)/tw.cfg          -> $(SEC_BIN) -i ;
  $(TWLKEY)/$(HOSTNAME)-local.key -> $(SEC_BIN) ;
  $(TWSKEY)/site.key       -> $(SEC_BIN) ;
}
```

```

#don't scan the individual reports
$(TWREPORT)                                -> $(SEC_CONFIG) (recurse=0) ;
}

# Commonly accessed directories that should remain static with regards to owner and
group
(
  rulename = "Invariant Directories",
  severity = $(SIG_MED)
)
{
  /                                -> $(SEC_INVARIANT) (recurse = 0) ;
  /home                            -> $(SEC_INVARIANT) (recurse = 0) ;
  /etc                             -> $(SEC_INVARIANT) (recurse = 0) ;
}
##### #
#
# File System and Disk Administration Programs
#
#####

(
  rulename = "File System and Disk Administration Programs",
  severity = $(SIG_HI)
)
{
  /sbin/accton                     -> $(SEC_CRIT) ;
  /sbin/badbblocks                 -> $(SEC_CRIT) ;
  /sbin/busybox                    -> $(SEC_CRIT) ;
}

##### #
# Kernel Administration Programs
#####

(
  rulename = "Kernel Administration Programs",
  severity = $(SIG_HI)
)
{
  /sbin/adjtimex                   -> $(SEC_CRIT) ;
  /sbin/ctrlaltdel                 -> $(SEC_CRIT) ;
}

##### #
# Critical devices # #
#####

(
  rulename = "Critical devices",
  severity = $(SIG_HI),
  recurse = false
)
{
  /dev/kmem                        -> $(Device) ;
  /dev/mem                         -> $(Device) ;
  /dev/null                        -> $(Device) ;
  /dev/zero                        -> $(Device) ;
}

```

## Appendix 2 – Sample Tripwire Report

```
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
Wrote report file:
/var/lib/tripwire/report/localhost.localdomain-20030210-165624.twr
```

### Tripwire(R) 2.3.0 Integrity Check Report

```
Report generated by:      root
Report created on:      Mon 10 Feb 2003 04:56:24 PM EST
Database last updated on: Never
```

```
=====
Report Summary:
=====
```

```
Host name:                localhost.localdomain
Host IP address:          127.0.0.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/localhost.localdomain.twd
Command line used:        tripwire --check
```

```
=====
Rule Summary:
=====
```

```
-----
Section: Unix File System
-----
```

Rule Name	Severity Level	Added	Removed	Modified
Invariant Directories				
Temporary directories				
* Tripwire Data Files				
Critical devices				
* User binaries				
Tripwire Binaries				
Critical configuration files				
Libraries				
Operating System Utilities				

```
66          0          0          0
33          0          0          0
100         1          0          0
100         0          0          0
```

66	1	0	1			
100	0	0	0			
100	0	0	0			
66	0	0	0			
100	0	0	0			
Critical system boot files	100			0	0	0
File System and Disk Administration Programs						
	100			0	0	0
Kernel Administration Programs	100			0	0	0
Networking Programs	100			0	0	0
System Administration Programs	100			0	0	0
Hardware and Device Control Programs						
	100			0	0	0
System Information Programs	100			0	0	0
Application Information Programs						
	100			0	0	0
Shell Related Programs	100			0	0	0
Critical Utility Sym-Links	100			0	0	0
Shell Binaries	100			0	0	0
System boot changes	100			0	0	0
OS executables and libraries	100			0	0	0
Security Control	100			0	0	0
Login Scripts	100			0	0	0
* Root config files	100			153	1	

17

Total objects scanned: 15585

Total violations found: 174

```

=====
Object Summary:
=====

```

```

-----
# Section: Unix File System
-----

```

```

-----
Rule Name: User binaries (/usr/sbin)
Severity Level: 66
-----

```

```

Added:
"/usr/sbin/t.txt"

```

```

Modified:
"/usr/sbin"

```

```

-----
Rule Name: Tripwire Data Files (/var/lib/tripwire)
Severity Level: 100
-----

```

```

Added:
"/var/lib/tripwire/localhost.localdomain.twd"

```

```

-----
Rule Name: Root config files (/root)
Severity Level: 100
-----

```

```

Added:
"/root/.mozilla/default/hars81zq.slt/Cache/9D62C735d01"

```

"/root/.mozilla/default/hars8lzq.slt/Cache/EB0CA7B6d01"

Modified:

"/root/.gconfd"

"/root/.gconfd/saved\_state"

"/root/.xsession-errors"

-----  
Rule Name: Root config files (/root/.gnome-desktop)

Severity Level: 100  
-----

Removed:

"/root/.gnome-desktop/Red Hat Linux-i386 8.0"

=====  
Error Report:  
=====

-----  
Section: Unix File System  
-----

1. File system error.  
Filename: /usr/sbin/fixrmtab  
No such file or directory

-----  
\*\*\* End of report \*\*\*

Tripwire 2.3 Portions copyright 2000 Tripwire, Inc. Tripwire is a registered trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY; for details use --version. This is free software which may be redistributed or modified only under certain conditions; see COPYING for details. All rights reserved.  
Integrity check complete.

**Product:** Tripwire Open Source, Linux Edition

**Product Type:** Utility

**Author:** Steve Fritts

---

### **Problem Background**

A couple of whitepapers have been written explaining how Tripwire can be used for protecting patient records in the healthcare industry, and for achieving certain parts of HIPAA compliance<sup>4</sup>. I will next take the knowledge I have gained from this project and see how it can be applied to my thesis, which is meeting HIPAA security standards. I believe this product can be used to meet some of the "auditing" requirements of the HIPAA Security Rule.

### **Product Placement**

Tripwire Open Source, Linux Edition is an open source version of commercial intrusion detection software from Tripwire Inc. It is designed for Linux software, and will run on several versions of Red Hat Linux. It is available as a free download from numerous websites, including [www.tripwire.org](http://www.tripwire.org) and [sourceforge.net](http://sourceforge.net).

Tripwire Open Source serves as a basic intrusion detection system. It looks for changes made to a computer's file system and logs any changes found, including information about file sizes and last modification dates. It can then write these changes to a log file, e-mail them to a system administrator, or send them to a printer.

After installation, I tested out many of the product's features, such as running integrity checks, configuring the product for automatic checks, saving reports, changing policy and configuration files, and sending reports automatically via e-mail.

I installed Tripwire Open Source on a Pentium II 400 computer with 128MB RAM running Linux Red Hat 8.0 with a standard 'server installation'. The Red Hat server installation I chose left out many packages which are listed in the default tripwire policy file, which had ramifications as discussed below.

Tripwire Open Source works by first making a "snapshot" of a Linux operating system's file structure, which is written to a special database. A current snapshot can then be taken, either at regular intervals or on a system administrator's whim. The current snapshot is compared with the initial database snapshot. Any changes found between the two snapshots are added to a report which is available to the system administrator.

### **Installation Overview**

Installation of Tripwire Open Source is straightforward. The software is available in RPM or tarball format--once the software is unpackaged, it must be compiled. Users can avoid having to compile the software by simply downloading and unpacking precompiled versions (which is

---

<sup>4</sup> 'Tripwire Software Protects Data and Network Integrity, Helps Healthcare Systems Meet HIPAA Privacy and Security Standards'. 2002. <[http://www.tripwire.com/files/literature/white\\_papers/Tripwire\\_21\\_CFR11.pdf](http://www.tripwire.com/files/literature/white_papers/Tripwire_21_CFR11.pdf)>; 'Tripwire and 21 CFR11 - Ensuring Integrity and Trustworthiness of Electronic Clinical Data'. 2002. <[http://www.tripwire.com/files/literature/white\\_papers/Tripwire\\_21\\_CFR11.pdf](http://www.tripwire.com/files/literature/white_papers/Tripwire_21_CFR11.pdf)>

what I chose to do). Several preliminary steps must be taken in order to enable Tripwire Open Source to perform its basic operations. The installation and configuration steps are outlined below. Step-by-step instructions on installing and setting up Tripwire Open Source can be found at Red Hat's website. I have included most of the steps below.

### Step 1

I downloaded Tripwire Open Source from [www.tripwire.org](http://www.tripwire.org). I downloaded both the Red Hat 7.x RPMs and the tarball versions.

Red Hat 7.x RPM: <http://www.tripwire.org/files/rpm4/tripwire-2.3-47.i386.tar.gz>

Open Source tarball: <http://www.tripwire.org/files/tripwire-2.3-47.bin.tar.gz>

### Step 2

I logged into my machine as "root" and installed the Tripwire RPM package.

### Step 3

I ran `twinstall.sh` to configure tripwire and set default password keys. Two key passphrases are set up when this script is initially run: a "site" and "local" key. Both keys are used to reconfigure the tripwire policy and configuration files later on.

### Step 4

I ran the tripwire program a few times just to see what happened. What became apparent almost immediately was that the default policy file used to create tripwire's database was largely different from the file structure I am running on my machine. The result of this is that numerous errors are listed each time tripwire is run.

### Step 5

I made backup copies of two important files: `twcfg.txt` and `twpol.txt`. `twpol.txt` contains all of the policies tripwire checks for when it is running. It is here that a user can modify directories and files that tripwire will check or ignore. `twcfg.txt` has some settings for general tripwire operation (such as where critical files needed to run tripwire are stored) that can be modified. Here is a look at my final `twcfg.txt` file:

```

ROOT                =/usr/sbin
POLFILE             =/etc/tripwire/tw.pol
DBFILE              =/var/lib/tripwire/$(HOSTNAME).tdw
REPORTFILE          =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE         =/etc/tripwire/site.key
LOCALKEYFILE        =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR              =/usr/bin/pico
LATEPROMPTING       =false
LOOSEDIRECTORYCHECKING =true
MAILNOVIOLATIONS    =true
EMAILREPORTLEVEL    =3
REPORTLEVEL         =3
MAILMETHOD          =SENDMAIL
SYSLOGREPORTING     =false
MAILPROGRAM         =/usr/sbin/sendmail -oi -t

```

The only changes I made here were to change `EDITOR` from `"/usr/bin/vi"` to `"/usr/bin/pico"`, which I am more comfortable with.

### Step 6

Next, I again ran the `twinstall.sh` script. This allowed me to enter site and local passphrases (again - I had forgotten my passwords and had to run this command again). After creating the passphrases, tripwire created binary files (`tw.cfg` and `tw.pol`) based on the settings in `twcfg.txt` and `twpol.txt`, respectively.

### Step 7

I then ran the command `tripwire --init` to allow tripwire to create an initial checksum database for my file system. This generated two files: `/var/lib/tripwire` and `/var/lib/tripwire/reports`.

### Step 8

Next, I began editing `twpol.txt` to create a new database entry that would not generate any missing file or directory errors. This took well over an hour. Once I was finished with this, I created a new `tw.pol` file with the command

```
/usr/sbin/twadmin --create-polfile -S site.key /etc/tripwire/twpol.txt
```

### Step 9

I deleted the dictionary file `/var/lib/tripwire/dhcpc2.twd`, then created a new dictionary file as before (with the command `tripwire --init`).

***NOTE:** Because it is relatively easy for an intruder to recreate a database snapshot by using the commands in steps 8 and 9 above, it is recommended that users delete the `twpol.txt` and `twcfg.txt` files after they are used.*

### Step 10

After recreating my database, I configured `twpol.txt` to send me e-mails automatically and changed the "HOSTNAME" field to match that of my computer (`dhcpc2`), and thus eliminating another error.

### Step 11

I tested e-mail transmission with the following command:

```
tripwire --test --email myEmail@earthlink.net
```

This worked perfectly. I also tested my database file with the following command:

```
tripwire --check
```

This instructed tripwire to compare my current file system image with the file system image stored in tripwire's database (a sample report is shown in Appendix A).

In order to protect Tripwire files (including the database and configuration files as well as any log files) from being modified by a hacker, it is important to verify file permissions or save them to removable media. It is not advisable to store all of these files on the same machine that Tripwire is monitoring.

### Lessons Learned/Problems

Running the command `tripwire --check` allows tripwire to check the current file configuration with what it has recorded in its database. A sample of a final tripwire report is included in this document (see Appendix A). This report is saved to a file, but can also be e-mailed to a system administrator or sent to a printer. Running the command `tripwire --update` is used to make changes to the database file. This is useful after you have made changes to the file system (for example, when software has been upgraded or new software installed). Checking my entire file system (as described in the `twpol.txt` file) with the `tripwire --check` command took my machine approximately 1.5 minutes to complete.

In order to protect Tripwire files (including the database and configuration files as well as any log files) from being modified by a hacker, it is important to verify file permissions or save them to removable media. It is not advisable to store all of these files on the same machine that Tripwire is monitoring.

### Final Results/Recommendations

Tripwire Open Source is a valuable intrusion detection tool that is easy to customize and run. Most of my time spent on this project (approximately 15 hours) was spent troubleshooting problems with the initial software installation and configuration. Once I overcame these hurdles, installing and configuring tripwire was quite easy.

Tripwire Open Source can be customized to meet a variety of needs. The `tripwire --check` command can be added to crontab, so that system administrators can automatically run tripwire to perform periodic checks of their file systems. Although it took over a minute for tripwire to check my entire file system, this time could be greatly reduced by reconfiguring the tripwire policy file to only check key directories and/or files. The time could be further reduced by running tripwire on more modern machines OR to have several instances of tripwire running on the same machine, with each instance checking a set of files. Having such a setup, and adding these tripwire instances to crontab where they could run checks every minute or so would give a system administrator frequent checks of the file systems of key machines. Tripwire Open Source is a great asset, both in terms of increased security and disaster recovery.

### References

- Fuller, Johnray. 'Red Hat Linux 8.0: The Official Red Hat Linux Reference Guide'. 2002. 3 Mar. 2003 <<http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/ref-guide/ch-tripwire.html>>
- Larkin, Meryll. 'How to Linux - Tripwire'. 29 Dec. 2002. 3 Mar. 2003 <<http://www.alwanza.com/howto/linux/tripwire.html>>
- McKeehan, Paula. 'Tripwire - An Integrity Assessment Tool'. 24 Jan. 2001. 3 Mar. 2003 <<http://www.sans.org/rr/audit/tripwire.php>>

## Appendix A: Sample Tripwire Open Source Report

```
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
Wrote report file: /var/lib/tripwire/report/dhcpc2-20030303-143417.twr
```

Tripwire(R) 2.3.0 Integrity Check Report

```
Report generated by:      root
Report created on:      Mon 03 Mar 2003 02:34:17 PM EST
Database last updated on: Never
```

```
=====
Report Summary:
=====
```

```
Host name:                dhcpc2
Host IP address:          Unknown IP
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/dhcpc2.twd
Command line used:        tripwire --check
```

```
=====
Rule Summary:
=====
```

```
-----
Section: Unix File System
-----
```

Rule Name	Severity Level	Added	Removed	Modified
* Tripwire Data Files	100	1	0	0
* Critical configuration files	100	0	0	2
* System boot changes	100	12	0	21
* Root config files	100	1	0	0

```
Total objects scanned: 15457
Total violations found: 37
```

```
=====
Object Summary:
=====
```

```
-----
Rule Name: Tripwire Data Files (/var/lib/tripwire)
Severity Level: 100
-----
```

```
Added:
"/var/lib/tripwire/dhcpc2.twd"
-----
```

```
Added:
"/var/log/wtmp.1"
"/var/log/honeyd"
-----
```

```
Modified:
"/var/log/boot.log"
"/var/log/cron"
"/var/log/maillog"
"/var/log/messages"
"/var/log/secure"
-----
```

```
-----
Rule Name: Critical configuration files (/etc/sysconfig)
Severity Level: 100
-----
```

# *Applications*

**Product:** Big Brother System and Network Monitor

**Product Type:** Utility for monitoring systems and networks.

**Author:** Amanda Hickman

---

### **Problem Background**

There are several different system and network monitors available in the market. Big Brother offers both a client and server version of its software. The client version only monitors one system such as the disk capacity and cpu usage. The server version monitors a network and the status of available services like DNS, FTP, and Telnet. This document describes the server version of the software. I became interested in this software after reading about it on a technical community I frequent as well as in some textbooks. I installed this software on both Linux and Windows to determine how easy it is to install and how useful the software actually is.

I have never really seen or used a network monitor before. In the past, I have used ETSU's Intermapper service to check on the status of servers, but it only gave the server name, not the services available, and the uptime/downtime. Therefore after reading about Big Brother, I became interested in giving it a try.

### **Product Placement**

Big Brother is a system and network monitor. It is mostly used for monitoring networks and network services. Big Brother monitors a variety of services including DNS, FTP, HTTP, SSH, POP3, and IMAP. It offers a simple web interface that is color coded to let the network administrator know about a problem.

There are two versions of Big Brother available. The company offers a free version for non-commercial use as well as a paid version for commercial use. In addition to the server version, a client version is offered for Windows machines that monitors disk and CPU usage.

### **Installation Overview**

*Note: the installation instructions are found in the README files.*

I installed Big Brother on both Windows and Linux. This section will give an installation overview of the installation on both systems. On either operating system it is crucial to have a web server already installed.

#### Windows Installation

The Windows installation requires that a web server already be configured. Any web server is permissible (Apache, IIS, Personal Web Server). I chose IIS since I already had it installed. The Windows installation is actually quite simple if IIS (or Personal Web Server) is already installed. The Windows installer of Big Brother creates all the virtual directories for you.

If you're using another web server like Apache, then the setup must be performed manually by following the installation instructions in the README file after installing Big Brother.

Use the default directory or choose another directory. Note that NT services sometimes

dislike to have spaces in its directory structure (i.e. **AVOID** using "Program Files" and such directories) since the spaces may pose a problem in web browsers. Do **not** install BBNTD on a FAT partition.

After the installation it is important to go to the Big Brother help page to get information on how to configure the `bb-hosts.cfg` file. The file should be configured as follows:

```
#summary bigbrother.bb 10.10.10.10 http://risserver.ris.com/bb/bb.html
#summary bigbrother.bb2 10.10.10.10 http://risserver.ris.com/bb/bb2.html

group-compress <h3>servers bb4</h3>
10.10.10.10 riserver # BBPAGER BBNET BBDISPLAY ftp http
10.10.10.20 install # ftp dns http imap pop3

summary bigbrother.bb 10.10.10.10 http://10.10.10.10/bb/bb.html
summary bigbrother.bb2 10.10.10.10 http://10.10.10.10/bb/bb2.html
```

### Linux Installation

The Linux installation requires a lot more configuration and setup than the Windows installation. The recommended web server for this install is Apache. The installation instructions for Big Brother are found in the README and INSTALL files after unzipping the Big Brother archive. Also, the `bb-hosts.cfg` file should be configured as above (in the Windows installation).

### **Using Big Brother**

Figure 3.1 below shows an example of Big Brother's main page. Services in green are up and running fine. Services in red mean trouble and are not available. Services in orange are still available, but need attention.

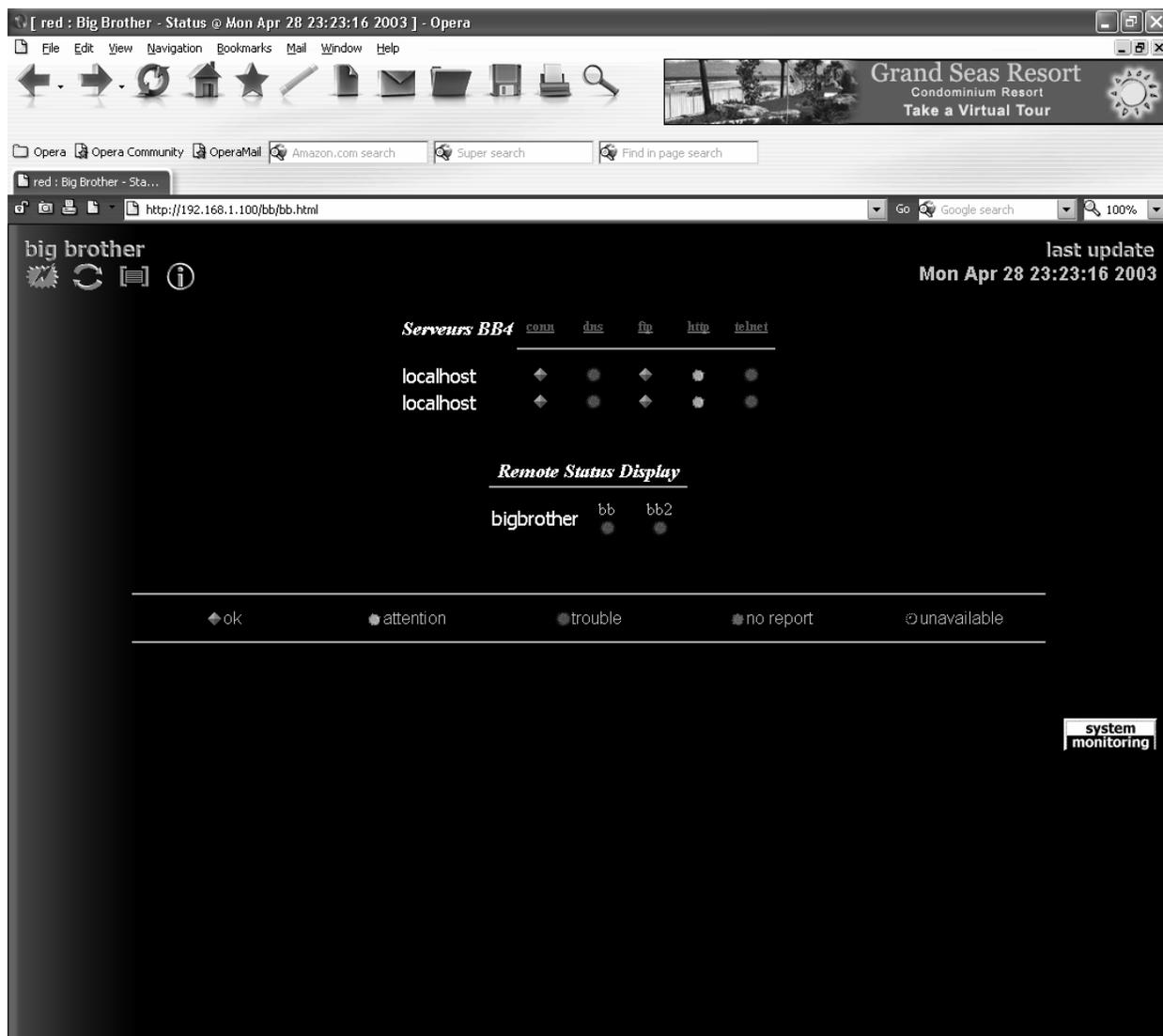


Figure 3.1: Big Brother main page

When you click on the flashing circle, you can gather information about the service. Figure 3.2 shows an ftp service, and it is fine. Big brother tries to ftp using port 21 and it is successful.

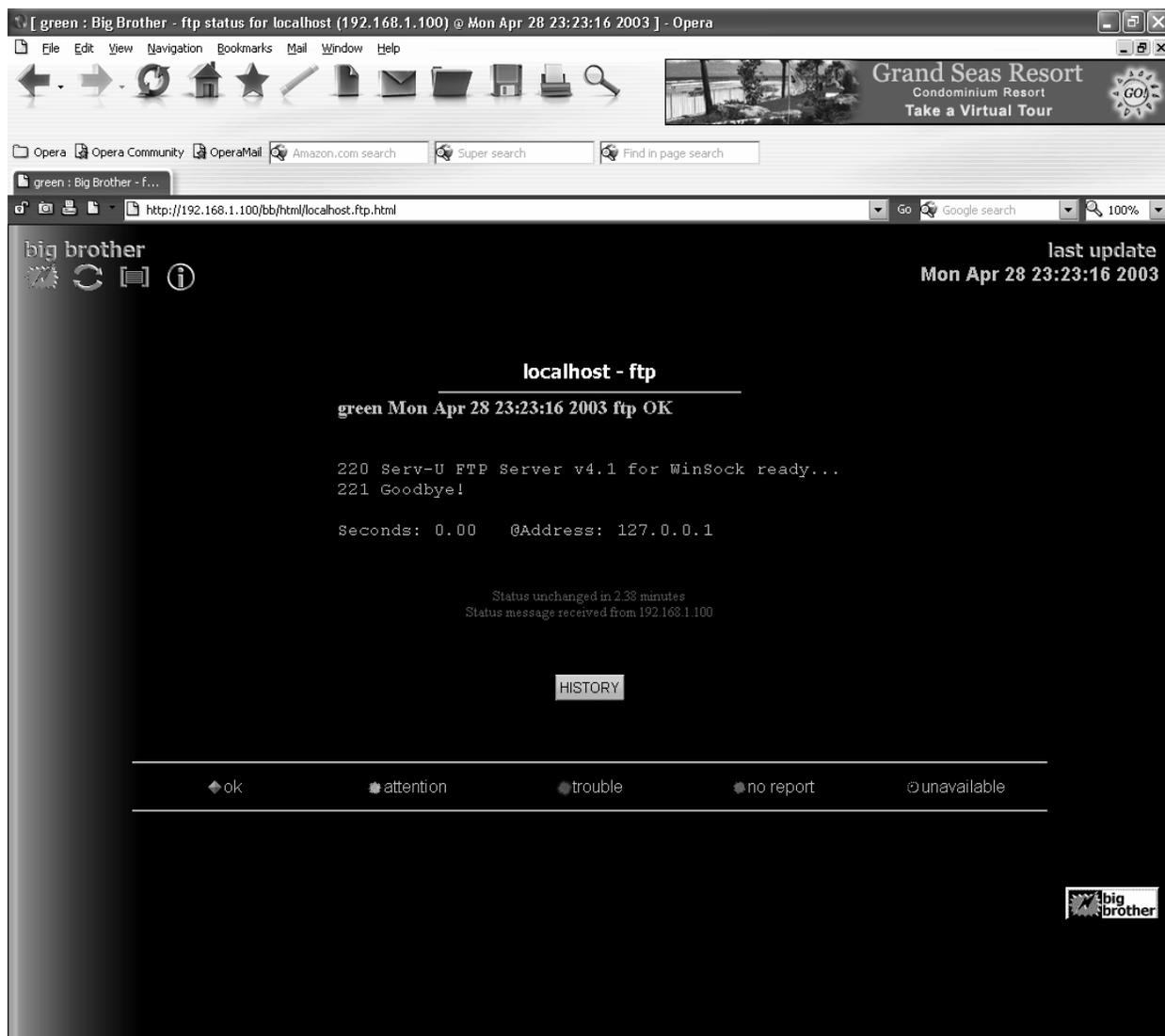


Figure 3.2: Big Brother FTP service

Figure 3.3 below show a service that is not working. Telnet is not setup on this machine. Therefore when Big Brother attempts to connect to the telnet port, it is unable to do so.

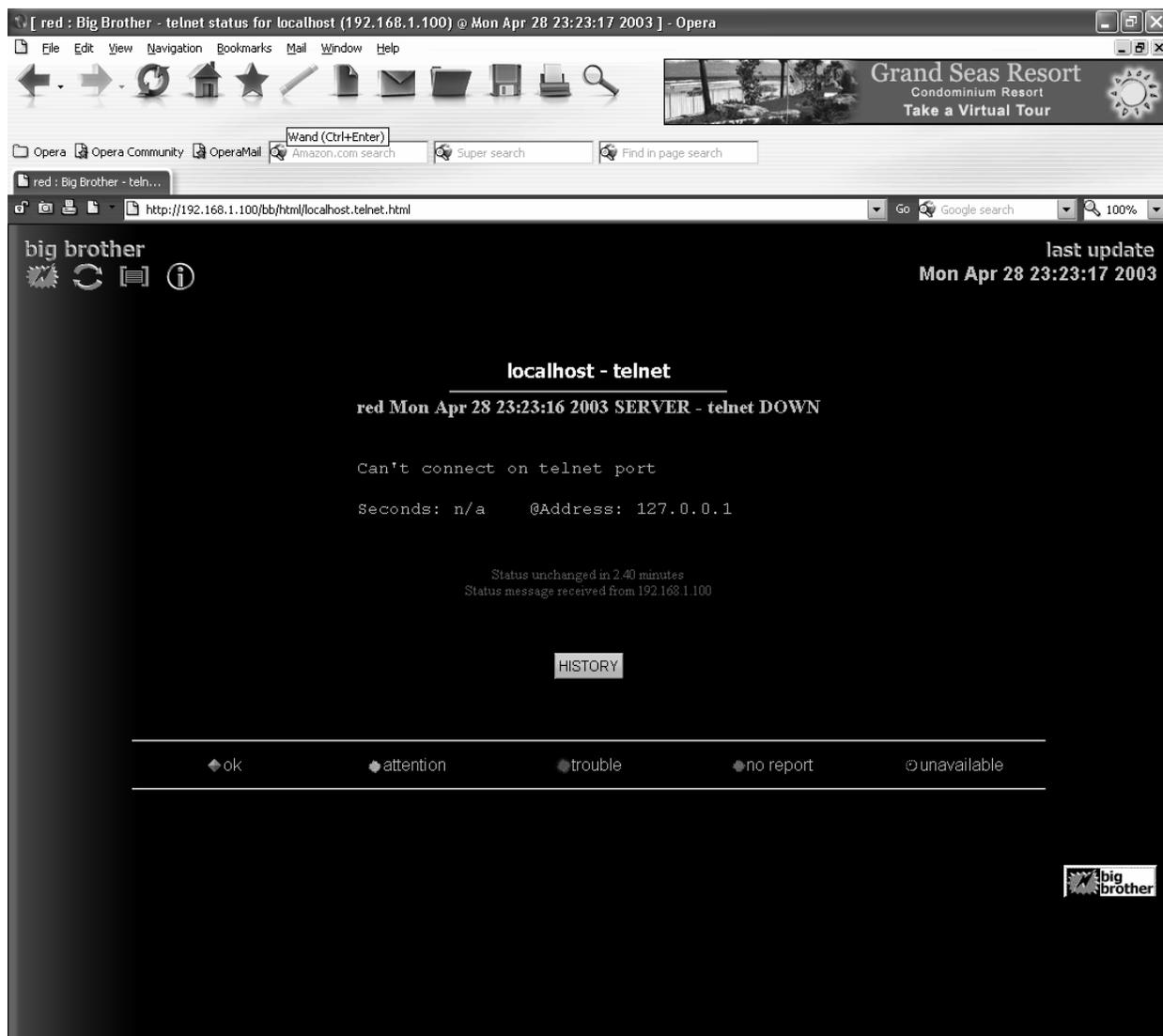


Figure 3.3: Big Brother attempts to connect to a service that is not available.

Figure 3.4 shows an example of a report for a single service. The report gives a percentage of how much the service has been available and not available. It also lists the days and times of when it was and was not available and for how long.

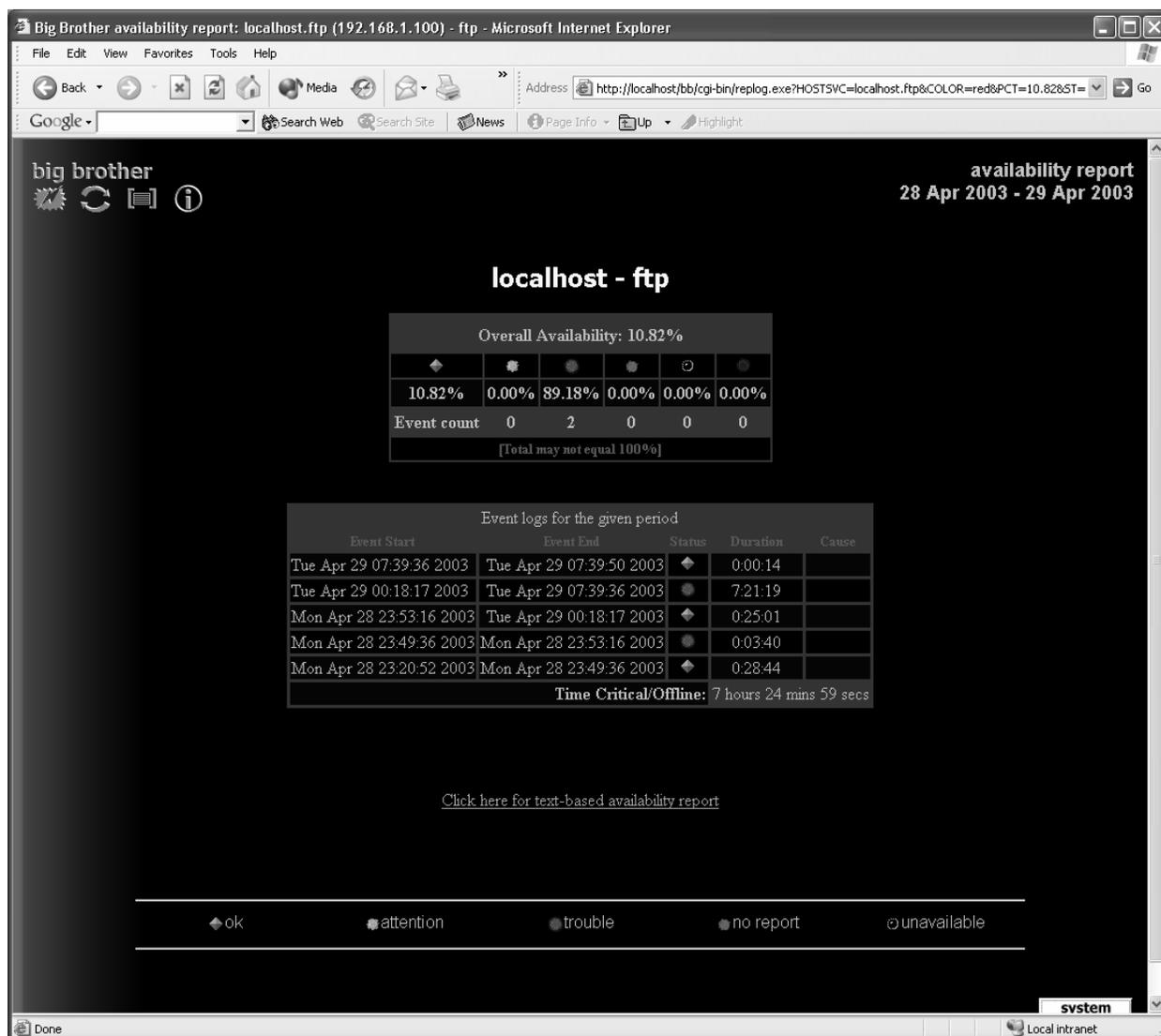


Figure 3.4: Services report

This is an example of a report for all the services. It is color-coded like the main Big Brother page. Green indicates 100% availability, while red indicates less than 97 percent availability. Also, you can see for the ftp service, that is it is in red and listed as being available roughly 11% of the time.

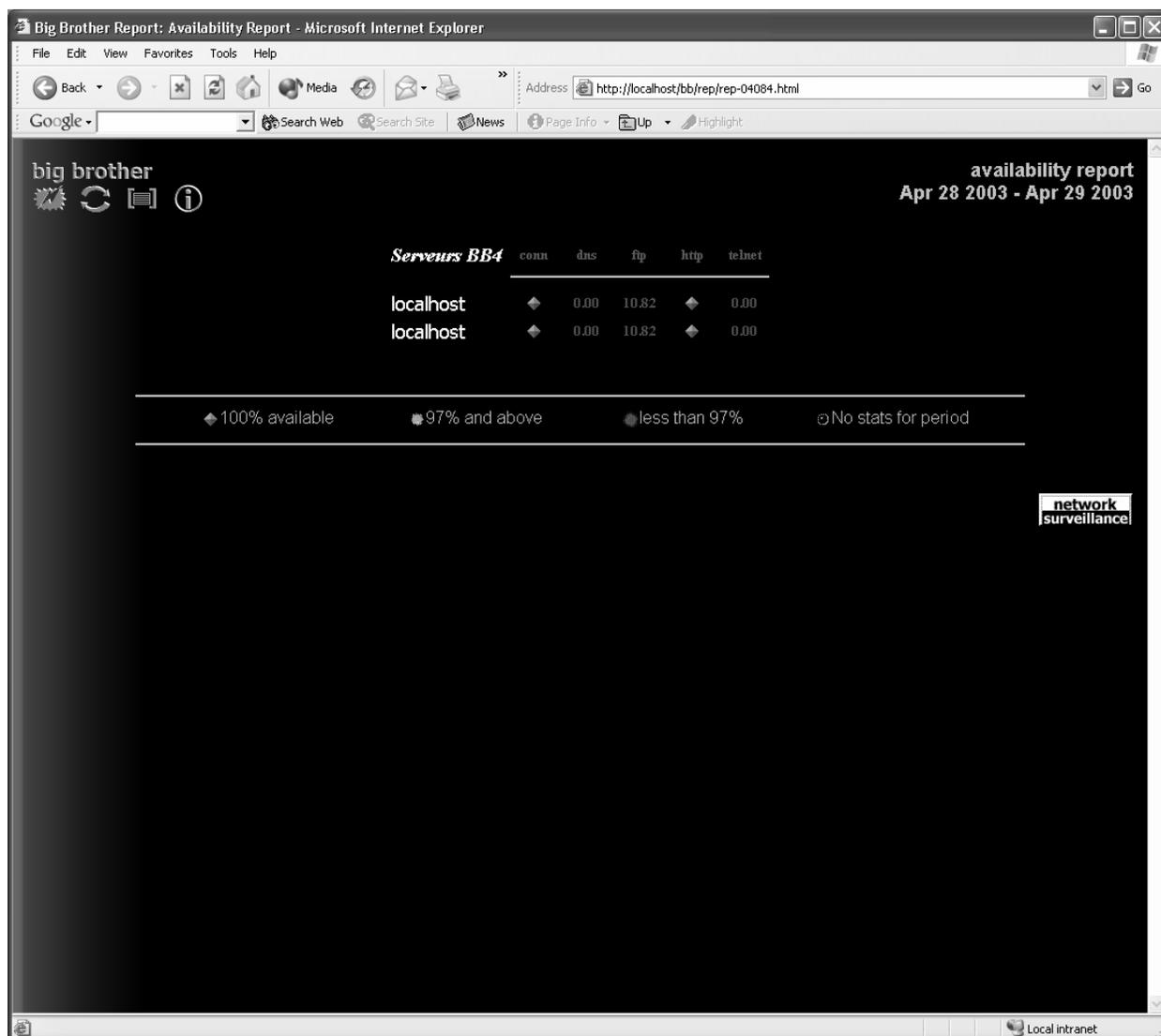


Figure 3.5

### Lessons Learned/Problems

Honestly, I did not run into any problems with the setup and configuration of Big Brother. Though the Linux installation was slightly more difficult, I was still able to get it up and running. The Windows installation was also not bad.

This project has taught me a lot about services and how to monitor them. I've also learned about setting up web servers on Linux and Windows which is something that I have never done before and have been quite interested in doing.

### Final Results/Recommendations

This is one of the coolest products I have installed this semester. I love the web interface and it is really easy to use. Big Brother also has a great reporting tool that allows you to view a report of the uptime/downtime of a particular service over a period of time as well as comparing it to the other services. I have included screenshots of Big Brother in the appendix of this document.

Big Brother provides several notification methods to notify system administrators when a problem occurs. Notification may be set up to notify based on time-of-day, specific machine, or test failure. The notification may be sent via email using SMTP, alphanumeric paging using Sendpage or Qpage, or numeric messages sent via SMS.

Something I did notice, however, is the time it takes for the web page to update itself after a service has been lost. You click on a service and it tells you it's down but the initial Big Brother web page was not updated. I realized that this value can be changed in the bbdef.cfg file. The default value is 300 seconds (5 minutes), but that is easily changed to whatever value desired.

Big Brother is quite customizable if using the Linux version. The Linux software download includes all of the source code, so notification schemes, color schemes, and plugins to other products (like bandwidth monitoring) may be added to suit a particular need. This is a big advantage I see over the Windows version which, on the other hand, only allows you to change color schemes and add plugins.

I found it useful for the e-mail server project as well as for my Windows Remote Installation project to monitor the status of services. This is definitely a tool I would want to have if I was a system administrator since it is easy to use and install and provides a valuable service.

### **References**

Big Brother. BB4. 2002. 30 April 2003. <<http://bb4.com/>>

**Product:** BrightStor ARCserve  
**Product Type:** Enterprise backup solution  
**Author:** Robert Nielsen

---

### **Problem Background**

If everything always went right, there would be no need for administrators to backup data. In the real world, things quite often don't go right. Though proper maintenance and configuration of software and hardware can reduce the frequency with which backups are needed, even the best of systems can experience failure. There is also the issue of systems that are working fine needing data restored because of user error. One of the places where users often need assistance in recovering lost data relates to E-Mail.

### **Product Placement**

It's easy to accidentally delete a message or to intentionally delete one only to realize later that the information from that message is still needed. There are also times when E-Mail must be archived for legal or corporate purposes. Because of situations like these, there is the need for a solid backup solution for use with E-Mail servers. Of course, for any backup solution to be useful, it must be able to not only handle the backup tasks well, but also restore quickly and easily too. In the case of E-Mail servers that are operational twenty four hours a day, seven days a week, the solution must also be able to work with the E-Mail server's databases while they are open and in use.

### **Installation Overview**

To understand better the products available and how they handle E-Mail server backups, two products were examined in a Microsoft Exchange environment. In this environment, Exchange was running on a domain controller for the domain "example". The backup hardware consisted of a Tandberg Data DLT7000 in an external enclosure and attached to a dedicated backup server, called "backup", via an Adapted AHA-2930CU SCSI controller. Like the domain controller, the backup server was running Windows 2000 Advanced Server as its operating system. Once attached to the system, the drive was immediately recognized by the SCSI controller. The operating system evidently does not include drivers for DLT tape drives though. Thus it was necessary to obtain drivers directly from Tandberg and install them so that the drive would work properly under Windows 2000.

To start the testing, a baseline was obtained by using Microsoft Windows Backup. This package is included in Windows 2000. It is essentially a pared down version of Backup Exec, a Veritas product. It can be started from the command line using `ntbackup.exe` or from the GUI by selecting Start, Programs, Accessories, System Tools, and Backup. Once started, selecting the Backup Wizard allows the user to select local or network-based data that they want to backup. It also allows a destination to be selected for the backup, including the ability to backup to file as well as to tape. There are only a handful of options available beyond the type of backup desired and the files to exclude.

Testing Windows Backup against the mailstore revealed the fact that Windows Backup, though supposedly Exchange aware, is not always. There were no options for Exchange listed at all. As

such, the folder containing the mailstore was selected for backup. Windows Backup was able to complete this backup of about 2.2 GB in about twenty five minutes. Once the backup was complete, testing was done on the system's ability to restore data. First, a restore to the local server was done. Using the local server instead of the remote one provided, by comparison, excellent restore times. The local restore only took about nine minutes to complete. Next a restore was done to the original storage location. This restore took about thirty three minutes to complete. Historically restores have been slower than backups, and in this case the restore time was about thirty three percent slower than the backup time.

To further investigate the ability of Windows Backup's compatibility with Exchange, the backup utility was started on the Exchange server. This version did offer additional support for Exchange that was not present in the version on the remote server. The assumption is that the Exchange installation either added functionality to the version of Windows Backup on this box or totally replaced the original version with an updated one. This version would allow the user to directly select Microsoft Exchange and then select from any of the available "Information Stores" on the server. The First Storage Group was selected and a backup created across the network to the backup server. Backing up like this would create a two step process that would require the tape drive to be used to make a backup of the Microsoft .bkf file which was created on the backup server. Obviously this would not be extremely efficient, but in some cases it may be better than having to do drive and folder based backups. If the version of Windows Backup on the backup server could have this functionality added, it would greatly improve its value as a backup solution.

With this testing complete, BrightStor ArcServe, a Computer Associates product, was installed on the backup server. The BrightStor Agent for Windows and Agent for Exchange were then installed on the Exchange server. ArcServe was able to locate and identify the tape drive without any issue. The interface for ArcServe was much more involved than the one provided by Windows Backup. It offered options for manipulating tapes, drives, media pools, etc. and also had options for generating reports as well. Using the backup wizard, a user can navigate through the network, individual servers, folders and files. For testing, the server MS was selected for backup. The wizard prompted for the media to backup to, asked what type of backup to run, asked if an agent should be used and finally asked for the user and password to use when connecting to the remote system. The whole feel of the package was nicer than that of Windows Backup. That is it was until the job status indicator popped up saying the job had failed. There was no obvious reason for the failure and the job logs did not indicate what had went wrong. The job was selected for modification and it was noted that though it was set as an Exchange backup, the fields related to Exchange were all blank. These fields were updated with the appropriate information related to the Exchange server and the job was resubmitted. It again failed. This time it reported an error, "EC=1219: failed to authenticate". Outlook Web Access was used to double check that the username and password being entered into the configuration were correct. The Exchange Client Agent configuration was started on the server and it offered the option of creating a new account for use by the backup software. The process was completed and the job restarted. Once again, the job failed. A search of the Computer Associates website yielded nothing related to the error code or the wording of the error. A generic web search located an article that mentioned the need for the ArcServe backup user to have particular rights. These included log on locally, log on as a service, and act as part of the operating system. The

domain policies were updated to match the settings listed and the job was restarted. With these settings in place, the job was able to backup the selected forty megabytes of data in less than twenty seconds! By comparison, this was about fifty percent faster than the backups had run under Windows Backup.

The next step for testing was to attempt a restore from ArcServe. The restore wizard was started and it prompted for the session to restore from and the location to restore the data to. The first restore attempt result in the job status indicator showing “crashed”. Once again there was nothing to indicate the cause of this failure and the logs were of no real use. The wizard was restarted and another restore job was created. For this restore job, the Exchange agent was turned off. This job also ended with a status of “crashed”. To see if other options were available besides what the wizard was presenting, it was decided that a restore job should be created manually. There were no substantial differences noticed in the settings available to a user creating a restore job manually versus one doing so with the restore wizard. There was a substantial difference in the job result though. The manually created job was able to complete the restore job in less than a minute.

One of the major differences found when using a \$1000 set of commercial packages versus one distributed for free with the operating system was ArcServe’s ability to do “brick” level backups on Exchange. Brick level backups essentially make backups of Exchange on a mailbox level instead of acting on the Information Store level. This gives the ability to backup and restore individual mailboxes directly. To test this functionality, several large E-Mails were generated and sent to one of the Exchange mail accounts. These messages totaled just less than three hundred megabytes. Once this was completed, a backup was made consisting of each of the mailboxes on the Exchange server. The total backup was about nine hundred and sixteen megabytes and took about nine and a half minutes to complete. The throughput here was about ninety five megabytes per second, noticeably slower than the Information Store based backup.

Once the backup was complete, a restore job was created. This job was set to restore the mailbox that had previously been sent the large messages. The restore configuration allowed this individual mailbox to be selected and the job took about nineteen minutes to complete the restore of almost three hundred megabytes. At one half megabyte per second, this restore was slow. Interestingly, it was still faster than the Information Store level restore by Windows Backup, which clocked in at only one seventh of a megabyte per second.

### **Lessons Learned/Problems**

The operating system evidently does not include drivers for DLT tape drives though. Thus it was necessary to obtain drivers directly from Tandberg and install them so that the drive would work properly under Windows 2000.

Arcserv had several issues like the job status indicator popped up saying the job had failed though there was no obvious reason for the failure and the job logs did not indicate what had went wrong. Other times it provided error messages like “EC=1219: failed to authenticate” or “crashed” without any real details to assist in correcting the problem.

**Final Results/Recommendations**

As expected, the more expensive commercial product offered more functionality than the free program. The issues found in the commercial product were somewhat unexpected though. Many of the error messages generated by the package were neither listed in the help files nor on the company's support website. The wizards appear to only be usable for a subset of the overall product functionality. Also, the configuration instructions seem to have left off several important steps needed to get things started. This combination of problems would make it hard to recommend this particular package. Based on the features, ArcServe did offer much beyond what the Windows Backup offered though. These features are probably enough that the software cost could easily be justified in an environment that required these additional features, assuming the individual configuring the software is highly familiar with it. For basic backup needs though, Windows Backup seems to be a better choice. Though a bit slower, it was much easier to configure and use making it a real bargain for the low end user.

**References**

- BrightStor ARCserve Backup. 2003. 23 April 2003. <<http://www3.ca.com/Solutions/ProductFamily.asp?ID=115>>
- Fugatt, Mark. "Backing up Exchange 2000 using Windows 2000 Backup". MSEXchange.org. 20 January 2003. 23 April 2003 <<http://www.msexchange.org/articles/MF020.html>>
- Microsoft Product Support Services. "XADM: White Paper - Disaster Recovery for Microsoft Exchange 2000 Server". 23 April 2003 <<http://support.microsoft.com/default.aspx?scid=kb;en-us;326052>>

**Product:** Bugzilla, Bugzero, Mantis, and PHPBugtracker

**Product Type:** Bug-Tracking Software

**Author:** Amanda Hickman

---

### **Problem Background**

I became interested in bug tracking software after installing and configuring Bugzilla on Linux and failing to install Bugzilla on Windows. I was curious if similar software exists that can be installed on both Linux and Windows. I also wanted to determine which software works better for which platform. I chose Bugzilla, Bugzero, Mantis and PHPBugtracker. All of these programs are used for bug tracking with which allow developers to track defects in software.

A bug-tracking system is basically a database of software bugs, bug resolution, customer complaints and hardware issues. Bug-tracking systems allow all of this to be centralized and combined into a single program rather than having everything written down in paper in a notebook or using several different software packages.

### **Product Placement**

Bugzilla, Mantis and PHPBugtracker are all open-source programs that fall in the category of “Bug-Tracking Systems” or “Defect Tracking Systems”. Bugzero is a commercial product which I obtained a free evaluation version. These products are mostly used by software vendors and open-source software developers to track bugs and issues related to software development.

The various software packages that I am evaluating developed out of some need. Bugzilla was developed to replace the internal bug-tracker for Netscape Communications. PHPBugTracker and Mantis have both been developed as a replacement for Bugzilla since they are easier to install and configure. Bugzero was developed to be a lightweight and easy to install commercial bug-tracking system.

### **Criteria**

When installing these software packages, I had several criteria in mind that I was looking for:

- Ease of Installation
- Working Installation on Linux
- Working Installation on Windows
- Documentation, Support Available
- Ease of Use
- Administration
- Customization

### **Installation Overview**

#### Bugzilla Installation for Linux and Windows

The Bugzilla website contains step-by-step instructions for installing and configuring the software on both Linux and Windows.

- The Linux version is found here: <http://www.bugzilla.org/docs216/html/stepbystep.html>
- The Windows version is found here: <http://www.bugzilla.org/docs216/html/win32.html>

Bugzilla may be obtained from here: <http://www.bugzilla.org/download.html>

In order for Bugzilla to be installed on either platform, the following components must be present:

- Web server (Apache recommended)
- Perl 5.005 (ActivPerl for Windows)
- A Database (MySQL 3.22.5 or higher recommended)
- Perl Modules which may be obtained from CPAN – the Comprehensive Perl Archive Network – <http://www.cpan.org>
  - Template (v2.07)
  - AppConfig (v1.52)
  - Text::Wrap (v2001.0131)
  - File::Spec (v0.8.2)
  - Data::Dumper (any)
  - DBD::mysql (v1.2209)
  - DBI (v1.13)
  - Date::Parse (any)
  - CGI::Carp (any)

Some of the Perl modules are difficult to install. It may be required to compile and install them yourself. It depends on which Linux distribution (RedHat, Mandrake) you are using and whether or not it comes with a package of the Perl modules. Also, RedHat now has a Bugzilla RPM which is used with the PostgreSQL database.

Also, always run checksetup.pl after installing each Perl module to ensure that it was installed correctly. The checksetup.pl file helps in determining problems with Bugzilla's installation. When the file is run and there are no problems, it creates a localconfig file which may then be edited to suit your installation.

The Linux instructions were extremely easy to follow, and if it hadn't been for the Perl module problems, I probably could have had Bugzilla running within an hour or two. The Windows instructions, however, were much more difficult to follow. They involve editing a lot of code and the installation is not very pretty. I would recommend not attempting to install Bugzilla on a Windows system.

#### Bugzero Installation for Linux and Windows

The installation for Bugzero on Linux and Windows is essentially the same. The installation instructions for Bugzero are found here: <http://www.websina.com/bugzero/install.html>; however, I have added some tips to make the installation somewhat smoother for a novice since it involves using the Java Standard Development Kit and an engine for Java servlets.

1. Obtain Bugzero from <http://www.websina.com/bugzero/>. This is a commercial software product and you must obtain a license to use it in business. Otherwise you may download the trial version.
2. Download and install Java SDK 1.3.1 or 1.4 from <http://java.sun.com>. *Do not download the JRE, make sure you download the SDK (Standard Development Kit)*. Add the

java\bin directory to the system path and then add an environment variable called JAVA\_HOME that points to the java directory.

3. Next, download and install Apache Tomcat from <http://www.apache.org>. To install Apache Tomcat, just unzip the file into the directory of your choice. This allows the web server to serve Java servlets and Java Server Pages (jsp). To test the installation, go to the bin directory Tomcat and type catalina run (catalina start on Windows). *You MUST have the java\bin directory in your system path and the JAVA\_HOME environment variable set in order to run Tomcat.* After launching tomcat type in <http://localhost:8080> in your web browser and you should get a default Tomcat page if your installation was successful.
4. MySQL is the recommended database for Bugzero. After installing and downloading it from <http://www.mysql.com> you need to do the following:
  - Create a database user account (bugzero) and an empty database (bugzero\_db)
 

```
mysql> CREATE DATABASE bugzero_db;
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, DROP ON bugzero_db * TO bugzero@localhost IDENTIFIED by 'bugzero_password';
```

*\*Note: bugzero\_password is the password you set for the bugzero account*

5. After performing all of the above steps, Bugzero is now ready to be installed. The entire Bugzero directory needs to be placed in Tomcat's webapps directory. Then launch the setup program under bugzero\WEB-INF. The setup program then guides the user through the rest of the installation, which is very short. Bugzero is then ready for use.

### Mantis Installation for Linux and Windows

The installation for Mantis is essentially the same for both Linux and Windows.

1. Download Mantis from <http://mantisbt.sourceforge.net/>
2. Ensure MySQL 3.23.3 or higher is installed
3. Ensure PHP 4.0.3 is installed
4. Install a Webserver (Apache, IIS)
5. Decompress the zipped archive and then rename it to "mantis" and move it to your web root

The Mantis website, has installation instructions that are easy to follow. The instructions are found here: <http://mantisbt.sourceforge.net/installation.php3>

### PHPBugTracker Installation for Linux and Windows

The installation of PHPBugTracker is essentially the same for both Linux and Windows. The installation instructions are found here: <http://phpbt.sourceforge.net/docs/installation.html>, but I believe my instructions are more thorough.

1. Ensure you are running a web server (Apache, IIS).
2. Ensure you have PHP 4.1.0 installed.
3. Ensure you have a database installed (MySQL recommended).
4. PEAR::DB which is the database abstraction layer that comes with PEAR in the recent versions of PHP.
5. Create a database for the software.
6. Download PHPBugTracker from <http://phpbt.sourceforge.net/>.
7. Decompress the archive and then copy it to your webroot.
8. After copying the directory structure, go to your web browser and navigate to the location where the directory was copied (<http://localhost/phpbug>).
9. A configuration page appears. The page will ask you for the root/administrator account for the database, the database name and some information for an administrator account.
10. After this information has been entered, submit the form and when you are asked, save the file it generates to the root directory of PHPBugTracker.
11. Now log in.

### **Product Analyses**

#### Bugzilla

Before I installed anything, I first read through all the documentation for the Bugzilla. I read through the Bugzilla Guide thoroughly before downloading Bugzilla or any of the required components.

While reading the documentation, I discovered that even though Bugzilla is designed for use on Linux systems, it has been successfully implemented on the Win32 platform as well. I first decided that I was going to attempt to install and configure it on my Windows XP machine. The documentation states that installing Bugzilla on Windows is “no picnic” and that is absolutely true. I first downloaded and installed MySQL and Apache HTTP Server. Then I installed all of the required Perl modules; however some would not install correctly. I then installed Bugzilla and made all of the required changes to the source code I spent roughly ten hours working on it and could never get it working quite right.

I then decided to install Bugzilla on Linux. I first installed Red Hat 8.0, then MySQL 3.23, and Apache HTTP Server 2.0. Afterward I installed Bugzilla in the `htdocs` directory of Apache. Once Bugzilla is installed, some Perl Modules must be installed. They available in a bundled format from CPAN and can be installed by typing

```
bash# perl -MCPAN -e 'install "Bundle::Bugzilla"'
```

After installing the Perl modules, the `checksetup.pl` script is run. This script checks the MySQL database install, Perl modules, and other configuration options to ensure they are consistent with Bugzilla's `.cgi` files. When I ran the script, it produced some errors relating to the Perl modules. I mostly had problems with the `DBD::MySQL` module. I spent around 12 hours trying to install the module. After searching through Google, I finally figured out how to make and install the

module myself. Afterward, I ran the `checksetup.pl` script again and it produced no errors and created all of my database tables for me.

I only had one other problem with my Bugzilla install on Linux. The problem was getting the Bugzilla start page to come up after starting Apache. I quickly figured out, by looking through the `localconfig` file, that I needed to define the webserver group to just to sets of double quotes. This is done if the install is just a test, which is what I was doing. After changing that one line, Bugzilla worked.

Once I had Bugzilla installed and running, I found it to be a pretty neat tool. I found it easy to create new accounts, change passwords, add bugs, search for bugs and configure. Bugzilla is easily configurable to your own liking by changing fields in the `localconfig` file as well as some of the other Perl scripts. Bugzilla templates which control the user interface may also be customized. This is done by editing Bugzilla's templates found in the `template` directory. Some templates that may be edited include:

- `index.html.tpl` – the main Bugzilla page
- `bug/create/user-message.html.tpl` – which tells the user how to report bugs
- `global/banner.html.tpl` – the banner that appears on all bugzilla pages.

Of course, there are more templates you can customize than just these. A more complete list of customizable templates and how to customize them is found in the Bugzilla documentation.

Overall, I found this software is extremely useful and easy to use once it has been installed.

### Bugzero

Bugzero is, by far, the easiest to set up. I had it set up and running in about 30 minutes. Since I am quite familiar with Apache Tomcat and Java, I was able to blaze through the installation. I would say that at most, it would probably take 2 hours to set up and configure if someone was unfamiliar.

The version of Bugzero I downloaded was a demo version. It came with 3 demo accounts (`dev`, `qa`, and `guest`) and a limit on how many bugs can be entered into the database. With these accounts, I was able to get a pretty good understanding of the software. I was able enter bugs and view reports with ease. The user interface is friendly and intuitive. It is not cluttered like I found Bugzilla to be. Bugzero also offers a very nice reporting system where you can select options for the reports. Bugzilla and PHPBugTracker also offer a reporting system, but these packages require you to download extra modules and do some additional configuration. Bugzero has this feature already built in.

Overall, I found Bugzero to be a very quick and easy install. It is a very small download and since it is written in Java it is lightweight and fast. The only drawback is the customization aspect. Since Bugzero is written in Java, it may be difficult to customize.

### Mantis

Mantis is a PHP/Web-based bug tracking system. It is still under development and available for beta testing.

I actually had a difficult time installing Mantis on both Linux and Windows. On Linux, I was never fully able to get Mantis to work. Mantis, for some reason, would not connect with the database on either system. I am unsure if it was a problem with Mantis itself or with my setup of Apache/MySQL/PHP. I searched through the Mantis documentation, as well as the bulletin board and found nothing that would solve my problem. I conducted a Google search and found one mention of my problem on a newsgroup, but there were no replies. I actually, re-installed Apache, PHP, and MySQL several times in attempts to get Mantis working, but never could.

I was finally able to get Mantis installed and working on Windows. I ended up installing IIS and PHP. Then I configured Mantis per the installation instructions and Mantis then was working. I am assuming that I must have configured Apache/PHP/MySQL in such a way that Mantis would not work.

Mantis is also easy to use once installed. One of the things I really liked about it, is that it has an administration web page that will test your installation and configuration of Mantis and indicate where problems lie. The administration web page also contains a link to another page that will allow you to customize Mantis. I thought this was great! It made customization very easy, however most of the customization centers around modifying the style sheets for fonts, tables, and colors. I searched through the documentation and could not find anything else about actually modifying the fields on the Mantis pages. I would assume that if someone took the initiative, they could modify Mantis's PHP files to get the customization they want.

Overall, though, I think Mantis has the potential to be a really great bug tracking system. However, it still needs a lot of work. As I mentioned earlier, it is still in the development phase. I would be interested in trying it again after a more stable release has been developed.

### PHPBugTracker

PHPBugTracker is a project that developed over frustrations installing and configuring Bugzilla. The developers hope that one day PHPBugTracker will be a replacement for Bugzilla. This software is by far my favorite of any of the bug tracking systems I have installed.

As with Mantis, I also ran into difficulty installing and configuring PHPBugTracker on both Linux and Windows. Initially, I installed the software and used Apache/PHP/MySQL. However, on Windows, I was able to complete the installation/configuration and log in, but I continually received a session variable error. PHPBugTracker provides links whenever errors occur. The link for the session variable error took me to the PHP development page and I found several solutions for the problem. Most of them involved ensuring that the session variable was pointing to C:\Windows\Temp. This was correct on my machine and I conducted several searches on Google and could not find anything of further help. My problem on Linux was somewhat different. I was never able to get past the installation/configuration page. The PHPBugTracker documentation, Google, and the PHP site offered no help with my problem. In

both cases, I am assuming the problem is similar to the Mantis problem. It must be an issue with the way I have configured Apache/PHP/MySQL.

However, I was able to successfully install and configure PHPBugTracker on Windows using IIS/PHP/MySQL. I really like the PHPBugTracker interface. It is simple, but professional looking. It's quite easy to use.

As for configuration, the config.php file allows you to change various database options and the CSS pages allow you to customize the colors and layout of the pages. As with Mantis, I could not find any documentation on actually changing the fields within PHPBugTracker. I would also assume here that someone could manually edit the PHP files and enter the appropriate information.

Overall, I think PHPBugTracker is a really nice software tool. It still needs some work to it, but it is at an almost stable release phase and I can easily see it taking the place of Bugzilla one day.

### Final Results/Recommendations

	<i>Installation</i>	<i>Linux</i>	<i>Windows</i>	<i>Documentation</i>	<i>User Friendliness</i>	<i>Administrat</i>
Bugzilla	Very Difficult	Yes	No	Excellent	Good	Good
Bugzero	Easy	Yes	Yes	Good	Good	Fair
Mantis	Easy	No	Yes	Poor	Fair	Good
PHPBugTracker	Easy	No	Yes	Fair	Good	Good

The above chart is a comparison of all four bug tracking systems. I rated the systems in the following ways:

The installation was rated from Easy to Very Difficult.

Was I able to create a working installation on Linux? On Windows?

The documentation is rated from Poor to Excellent.

User Friendliness is rated from Poor to Excellent.

Administration is rated from Poor to Excellent.

Customization is rated from Poor to Excellent.

Overall, I would probably choose Bugzilla as my bug tracking system of choice. Even though it took a good 30 hours to install and configure, especially with the Perl modules, it is an awesome bug tracking system. It has been around longer than any of the other systems and the documentation is quit extensive.

My second choice would be Bugzero since it is the easiest to install. The only drawback I see is that it is a commercial system and therefore it costs money for licensing. A full license is around \$1000 for unlimited use and 6 months of free support.

I don't plan on using any of these software packages in the near future. However, should I need one, I will be looking into one of them. I'm also planning on keeping my eye on PHPBugTracker and Mantis because I think they have real potential to replace Bugzilla.

**References**

Bugzero. WEBSina. 30 April 2003 <<http://www.websina.com/bugzero/>>

Bugzilla - Bug Tracking System. The Mozilla Organization. 25 April 2003. 30 April 2003  
<<http://www.bugzilla.org>>

Curts, Benjamin. PHPBugTracker. 30 April 2003 <<http://phpbt.sourceforge.net/>>

Mantis. 9 October 2002. 30 April 2003 <<http://mantisbt.sourceforge.net/>>

**Product:** Bugzilla  
**Product Type:** Bug Tracking System  
**Author:** Amanda Hickman

---

### **Problem Background**

I decided to install Bugzilla because I was interested in seeing how a bug tracking system worked. I've never used one before. In fact, I've never even set up a database or a web server, and this project gave me the opportunity to do both, as well as try out an interesting product.

### **Product Placement**

Bugzilla is a free, open-source program that falls in the category of "Bug-Tracking Systems" or "Defect Tracking Systems". The product may be used to by help desks to keep track of trouble tickets or by software vendors to track bugs and issues related to software development. It is comparable to free, open-source programs such as Bugzero, Gnats, and phpBugTracker. This is a product evolution of the latest stable release of Bugzilla v. 2.16.2, in which I discuss the ease of installation, configuration, and how well the product performed.

Bugzilla was originally written to replace the defect-tracking tool used by Netscape Communications. Later, Bugzilla was re-written in Perl and put to use by the open-source browser project, Mozilla. Today, it is one of the more popular open-source bug-tracking tools.

### **Installation Overview**

The Bugzilla website, <http://www.bugzilla.org>, has instructions for installing Bugzilla on a variety of platforms including Linux, Windows and MacOS. It is important to note, that in order for Bugzilla to work correctly, you must have Perl, MySQL database, and Apache webserver installed and working.

Before I installed anything, I first read through all the documentation for the Bugzilla. I read through the Bugzilla Guide, thoroughly before downloading Bugzilla or any of the required components. While reading the documentation, I discovered that even though Bugzilla is designed for use on Linux systems, it has been successfully implemented on the Win32 platform as well. I first decided that I was going to attempt to install and configure it on my Windows XP machine. The documentation states that installing Bugzilla on Windows is "no picnic" and that is absolutely true. I first downloaded and installed MySQL and Apache HTTP Server. Then I installed all of the required Perl modules; however some would not install correctly. I then installed Bugzilla and made all of the required changes to the source code I spent roughly ten hours working on it and could never get it working quite right.

I then decided to install Bugzilla on Linux. I first installed Red Hat 8.0, then MySQL 3.23, and Apache HTTP Server 2.0. Afterward I installed Bugzilla in the `htdocs` directory of Apache. Once Bugzilla is installed, some Perl Modules must be installed. They available in a bundled format from CPAN and can be installed by typing

```
bash# perl -MCPAN -e 'install "Bundle::Bugzilla"'
```

After installing the Perl modules, the checksetup.pl script is run. This script checks the MySQL database install, Perl modules, and other configuration options to ensure they are consistent with Bugzilla's .cgi files. When I ran the script, it produced some errors relating to the Perl modules. I mostly had problems with the DBD::MySQL module. I spent around 12 hours trying to install the module. After searching through Google, I finally figured out how to make and install the module myself. Afterward, I ran the checksetup.pl script again and it produced no errors and created all of my database tables for me.

### **Lessons Learned/Problems**

The only major problems I had with Bugzilla involved the Perl modules. There are so many different versions of each module, it's difficult to find the correct one. Once the correct one is found, it can be difficult to install because you may end up having to compile it yourself. I only had one other problem with my Bugzilla install on Linux. The problem was getting the Bugzilla start page to come up after starting Apache. I quickly figured out, by looking through the localconfig file, that I needed to define the webserver group to just to sets of double quotes. This is done if the install is just a test, which is what I was doing. After changing that one line, Bugzilla worked.

As for lessons learned, Bugzilla taught me a lot, mostly about Linux and Perl. I found this to be a very rewarding project.

### **Final Results/Recommendations**

Once I had Bugzilla installed and running, I found it to be a pretty neat tool. I found it easy to create new accounts, change passwords, add bugs, search for bugs and configure. Bugzilla is easily configurable to your own liking by changing fields in the localconfig file as well as some of the other Perl scripts. This software is extremely useful, and easy to use once it has been installed.

### **References**

"Bugzilla Bug Tracking System". 30 April 2003 <<http://www.bugzilla.org> >

**Product:** Concurrent Version System (CVS)

**Product Type:** Version Control Software

**Author:** C. Judith Nyabando

---

### Problem Background

The purpose of this project was to learn how to install and configure Concurrent Version System (CVS), and to configure remote access to CVS repository through Secure Shell (SSH).

### Product Placement

CVS is a version control software usually used by developers to keep track of the history of source files. Instead of storing files in the traditional way, CVS uses less storage space by storing all the versions of a file in a single file. This is done by storing differences between versions in the file. CVS stores files in a CVS repository and users access and manage these files through CVS commands.

### Installation Overview

CVS can be downloaded from a number of websites one of which is the CVS home page, <http://www.cvshome.org>. CVS also comes with Linux Server. For this project I had to install Red Hat Linux Server so there was no need to download CVS.

First of all I created two groups called `devproj` and `cvs`, and then I create a CVS root account, `cvsroot`, which would host the CVS repository and the CVS administrative files. I also created three ordinary user accounts: `cvsuser1`, `cvsuser2`, and `cvsuser3`. I added these users to the `devproj` and `cvs` groups.

The following command created the CVS Repository in `/home/cvsroot`:

```
cvs -d /home/cvsroot init
```

This command creates a repository and a directory called `CVSROOT`. `CVSROOT` contains cvs administrative files.

### Using CVS

There are several commands that users can use to get access to the CVS repository. In this section I discuss how I started a project, checkout a module, and check in a module. First of all I set up the `CVSROOT` environment variable in `.profile`:

```
CVSROOT=/home/cvsroot/  
Export CVSROOT
```

I then created a directory called `devproj` in `cvsroot` home directory and set up read, write and execute permission on this directory for the owner and group members. As `cvsuser1` I started a new project. I had a directory called `devproj` and it had a single file `fun.txt` on `cvsuser1` home directory. To put this directory in the repository I executed:

```
cvs import -m "the beginning" devproj cvsuser1 start
```

where `import` is the `cv`s command to import a module or directory into the repository, `-m` is the command to attach commit notes. This makes `fun.txt` available to the other users for checkout. So as `cv`suser2 I checked out `devproj`:

```
cv
```

s checkout devpoj

where `checkout` is the command to checkout a module for modifications. To check in the module:

```
cv
```

s commit -m devproj

The `-m` command is used to enter a message to avoid starting the editor. Every time a project is checked in the user must enter a record of the changes they made to the files in the module. So the user can use the `-m` option or the editor is started for them to enter the record.

The `history` command is used to show commit messages associated with a file. The `diff` command is used to show the differences between versions.

### Accessing CVS Repository Through SSH

A CVS repository can be accessed remotely i.e. the repository resides on one machine and users can keep a working copy of a module on another machine. The users can access the repository from their local machines in a client/server way. Usually this is done through remote shell (`rsh`) protocol. For this project, however I had to use the `ssh` protocol. `Ssh` is a more secure protocol that allows users to start a login shell on a server machine remotely. All data transmitted through `ssh` is encrypted unlike `telnet` where data is transmitted in plain text.<sup>2</sup> The goal for this project was to use `ssh` to execute `cv`s commands on the remote machine. `Ssh` comes with Linux but it can also be downloaded from several websites.

### Using ssh

The first thing I had to do was to set up the client and server machines on a network. Once network was configured I set the `ssh` environment variable in `.bash_profile`:

```
CVS_RSH=ssh
export CVS_RSH
```

This way the user does not have to enter the command `export CVS_RSH=ssh` every time they want to access CVS remotely. At this point the system is set up for remote CVS access. The following are the procedures for checking out and checking in `cv`s modules remotely.

To access the repository to check out a module a user will execute:

```
cv
```

s -d :ext:cvsuser1@cvshost:/home/cvsroot checkout devproj

where `cv`suser1 is the user name, `cv`shost is the name of the server – the remote machine that host the repository, `checkout` is the `cv`s command to check out a module, `devproj` is the module to be checked out.

The user will be prompted for a password. If the password is valid the module is checked out successfully. The user can work on the module locally. When the user is ready to check back the file into the repository they type the following command:

```
cvs -d :ext:cvsuser1@cvshost:/home/cvsroot checkin devproj
```

When releasing or checking in a module, the user will be prompted for the password twice.

### **Lessons Learned/Problems**

When I was setting up the network I ran into the problem that I had installed a medium firewall during the installation of the Linux Server. The firewall blocked or rejected ssh communication between the nodes on the network. To solve this problem I had to modify the ip chains to accept ssh communication.

### **Recommendations**

CVS is a good system to use for tracking file versions because it utilizes less storage space as compared to regular directories. It also minimizes human errors like deleting files by mistake. Users have restricted access to the repository through CVS commands and most of the commands always give the user an opportunity to verify the commands to be executed. For instance when a module is checked into the repository the user is asked whether the commit was successful or not. If not, the module will not be checked in.

Accessing cvs through ssh is good for the following reasons:

- It gives users remote access to the repository
- It provides a secure connect to a remote machine

Remote access to a CVS repository is particularly important if a user needs to access more than one repository located on different machine. For instance, a student might want to access repositories on Zephy, Einstein and Origin. Instead of going to the machines physically in order to access the repository they can sit in front of one machine and access all of these machines without much physical activity

### **References**

Price, Derek Robert. "CVS-Concurrent Versions System v1.11.3". 29 27 December 2002.

April 2003 <<http://www.cvshome.org/docs/manual/cvs.html>>

"Notes on Installing CVS". 29 April 2003 <<http://www.cs.princeton.edu/~wtcorrea/notes/cvs/cvs-notes.html>>

"SSH Tutorial for Linux". 29 April 2003 <<http://www.suso.org/linux/tutorials/ssh.phtml>>

**Product:** Light weight Directory Access Protocol (LDAP)

**Product Type:** (RedHat8.0, Maintain User information)

**Author:** Sai Divvala

---

### **Problem Background**

Lightweight directory access protocol (LDAP) is used to access X.500 directory services. LDAP protocol architecture consists of a Client-Server communicating via the TCP/IP networking model. LDAP information directory is a type of database which stores information about entries like people, machines, and offices. LDAP information directory is optimized for read performance. LDAP is not suitable when there are frequent data updates. Basically LDAP directory stores information that can be described in attributes. The LDAP information directory is hierarchical. Directory information tree is made up by the data present in the LDAP servers. Clients forward their query to the server and the server provides the client with the answer or provides the location of the requested data. LDAP provides global directory service i.e. clients view the same data on any LDAP server. Applications need not care about the type of the server that hosts the directory service as LDAP protocol is cross-platform and standard based. LDAP directory service replicates its data on many locations to improve the query response time and security. LDAP provides security by providing users with access control lists to perform operations on their data.

The basic information object of LDAP server is entry. Each entry's type is defined by an object class. An object class defines attributes. An entry is a collection of attributes. Each entry has a globally-unique distinguished name (DN), with which the entry is referenced unambiguously. Each distinguished name has a relative distinguished name made up of the entry's attributes. Name syntax of the LDAP directory entries begin at entry level and represent each level up to the top level. Successful search operation can be accomplished by providing the correct LDAP name format. Each attribute has a type and values. The values of attributes depend on the type of the attributes. Type "cn" indicates common name and its possible value is "Edward Thomas". The possible value of the attribute "mail" is Edward@mail.com.

### **Product Placement**

LDAP software is used for maintained the user information like, e-mail address, telephone number, address, etc. This can be used for the tracking the user information efficiently. LDAP provides e-mail address functionality to e-mail clients. LDAP server can also be used as authentication server with when it was integrated with PAM.

### **Installation Overview**

I downloaded the LDAP software files from [www.openldap.org](http://www.openldap.org) in the zip format. I logged into the system as root and checked for the presence of prerequisites like transport library services, kerberos authentication services, Berkley database software, POSIX threads and TCP wrappers. I unzipped the files into a previously created directory openldap. Then I ran the configure script and accepted the default values set by the configure script as the openldap documentation said that configure script normally auto-detect the appropriate settings. The settings include making the prerequisite software available to the LDAP server by looking at their location. Configure ran successfully without any problems. Then I built the dependencies by typing the command "make

depend”. Then I compiled the OpenLdap software by typing the command “make” which actually builds LDAP libraries. Then I tested to check if the software was properly configured by typing “make test”. Then I installed the software by typing the command “su root -c make install” and supplied the password. I checked if the installation process went correctly by verifying the presence of configuration file slapd and slurpd in /usr/local/etc/openldap. After installation I configured the slapd file. Slapd file consists of three types of configuration. They are global, backend, and database specific. Global information is specified first which is followed by backend information and then the database information. Global directives can be overridden in the backend directives which can be overridden in database directives.

My configuration file:

```
##### Global directives#####

include /usr/local/etc/schema/core.schema

# the above line includes core.schema configuration file which contain core
schemas.

access to * by * read

# above line indicates that read access is granted to all users at global level

#####Backend directives#####

3.database bdb
#above line indicates that LDAP server uses Berkley database
    suffix dc=wwldap, dc=edu

    # above line specifies the suffix for queries to pass to bdb database.

    directory /usr/local/var/openldap-data
    #above line specifies the directory in which the database files reside.
        rootdn "cn=Manager,dc=example,dc=com, mail=edwards@mail.com"
        rootpw secret
# the above two lines specifies the super-user id and password
index uid pres,eq
index cn,sn,uid pres,eq,approx,sub
index objectClass eq

#above line specifies the indices for various attributes
access to attr=userPassword
        by self write
        by anonymous auth
        by dn.base="cn=Admin,dc=wwldap,dc=edu" write
        by * none
    access to *
        by self write
        by dn.base="cn=Admin,dc=wwldap,dc=edu" write
        by * read

#above lines specify access controls for the entry Admin.
#####Database directives#####
# BDB definition for example.net
    database bdb
    suffix "dc=wwldap,dc=edu"
    directory /usr/local/var/openldap-data
    rootdn "cn=Manager,dc=wwldap,dc=edu"
```

```

index objectClass eq
access to * by users read
# I didn't use another database. So I did not override the database directives.

```

Then I edited the LDAP data interchange format (LDIF) by placing the user name and the object class:

```

Dn: cn= manager , dc= wwpdb , dc=edu
Objectclass=person

```

After editing the `Ldif` file and `slapd` file, I was able to search the “manager” e-mail id.

### Lessons Learned/Problems

The first problem I faced during the installation of LDAP is incompatibility issues of the packages with the operating system (OS) I am using. After reading through the entire documentation, I came to know that previously downloaded packages are not compatible with the OS I am using. Then I downloaded the right set of packages that are compatible to my OS.

From the above problem, I learnt that we should not completely depend on site that is providing the packages. We should read the documentation carefully and then decide on downloading the correct packages.

The second problem I faced due to the improper editing of LDAP application files (`slapd` and `Ldif`). Then I went through the documentation again and understood the meaning of all the lines in the file and then corrected my files. From the above problem, I learnt that we should understand content of the files to be modified and then proceed with the modification of files that suits our scenario.

### Final Results/Recommendations

LDAP can be used in the Wilson Wallace lab to maintain the list of users whose information can be as attributes like usernames, passwords, course id, e-mail ids and platforms on which the users are working. LDAP simplifies the change for users and also reduces the chance of having infrequently used accounts with forgotten passwords.

### References

OpenLDAP Foundation. 23 April 2003. 30 April 2003 <[www.openldap.org](http://www.openldap.org)>

**Product:** MySQL, Apache, PHP Installation for Linux

**Product Type:** Application

**Author:** Adam Berry

---

### **Problem Background**

This guide arose out of the lack of documentation I could find pertaining to current releases MySQL, Apache, and PHP. I searched exhaustively and consulted quite a few books and websites while putting this together. The following is a comprehensive tutorial that covers in detail Linux source installations of MySQL, Apache, and PHP under the most current releases as of 04/27/03.

The intended audience of this document is a beginner to source installations under Linux. The ability to create and modify files with a text-editor is necessary to complete this How-To.

### **Project Goals**

By using this document, you should be able to successfully install all three applications and have a working configuration in less than an hour, provided you can read and type at a moderate pace. My goal in writing was to spare others the 3 week ordeal I endured attempting to successful install the three.

The document is laid out in the following manner; commands that need to be entered in a terminal are in bold, an explanation of what the command is doing precedes in normal text. I have also provided an Overview for the Impatient which simply lists all commands, with no explanations for the advanced user.

Software:

- Red Hat Linux 9 (clean install)
- httpd 2.0.45 (SOURCE)
- MySQL 4.0.12 (SOURCE)
- PHP 4.3.1 (SOURCE)

Downloads:

- Grab the latest versions of MySql, Apache, and PHP from the following locations.

Current stable releases as of 04/15/03:

- httpd 2.0.45 (source) –  
<http://apache.mirrorcentral.com/dist/httpd/httpd-2.0.45.tar.gz>
- MySQL 4.0.12 (source) –  
<http://www.mysql.com/downloads/download.php>
- PHP 4.3.1 (source) –  
<http://www.php.net/downloads>

## Walkthrough

### Overview for the Impatient – MySQL

Before you do anything, create a user mysql in group mysql.

```
*****
* STOP *
*****
```

Make sure you have created the mysql user in group mysql.

#### # EXTRACT AND INSTALL

```
# su
# cd /home/your-user-name-here
# tar xfz mysql-standard-4.0.12-pc-linux-i686.tar.gz
# cd /usr/local/
# ln -s /home/user-name/mysql-standard-4.0.12-pc-linux-i686 mysql
# cd ./mysql
# scripts/mysql_install_db
```

#### # CHANGE OWNERSHIP

```
# cd /usr/local/mysql
# chown -R mysql data
# chgrp -R mysql .
# chmod -R go-rwx data
```

#### # LAUNCH SERVER

```
# cd /usr/local/mysql
# bin/safe_mysqld --user=mysql &
# bin/mysqladmin -u root status
```

#### # CREATE my.cnf FILE

```
# cd /etc
# touch my.cnf
```

Open the file with a text editor and insert the following two lines:

```
[mysqld]
user=mysql
```

Write the file.

#### # SET DAEMON TO START AT BOOTUP

```
# cd /usr/local/mysql
# bin/mysqladmin -u root shutdown
# chmod u+x support-files/mysql.server
# support-files/mysql.server start
# bin/mysqladmin -u root status
```

```

# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/
# cd /etc/init.d
# chmod 755 mysql.server
# cd /etc/rc2.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc3.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc5.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc0.d
# ln -s ../init.d/mysql.server K01mysql

```

## # SET SYMBOLIC LINKS TO COMMON TASKS

```

# ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql
# ln -s /usr/local/mysql/bin/mysqladmin /usr/local/bin/mysqladmin
# ln -s /usr/local/mysql/bin/mysqldump /usr/local/bin/mysqldump

```

## Overview for the Impatient – Apache

### # EXTRACT AND INSTALL

```

# su
# cd /home/your-user-name-here/
# tar xzf httpd-2.0.45.tar.gz
# ln -s httpd-2.0.45 httpd
# cd ./httpd

```

### # CONFIGURE AND INSTALL

```

# ./configure --prefix=/usr/local/apache \
--enable-so \
--enable-cgi \
--enable-info \
--enable-rewrite \
--enable-speling \
--enable-usertrack

# make
# make install

```

### # START DAEMON FOR FIRST TIME

```

# cd /usr/local/apache
# bin/apachectl start

```

### # SET DAEMON TO START AT BOOTUP

```

# cp /usr/local/apache/bin/apachectl /etc/init.d/httpd
# cd /etc/init.d
# chmod 755 httpd
# cd /etc/rc2.d
# ln -s ../init.d/httpd S99httpd

```

```
# cd /etc/rc3.d
# ln -s ../init.d/httpd S99httpd
# cd /etc/rc5.d
# ln -s ../init.d/httpd S99httpd
# cd /etc/rc0.d
# ln -s ../init.d/httpd S99httpd
```

#### # STOP DAEMON TO CONFIGURE PHP

```
# cd /usr/local/apache/
# bin/apachectl stop
```

#### # CREATE SYMBOLIC LINK TO START/STOP DAEMON FROM ANYWHERE

```
# ln -s /usr/local/apache/bin/apachectl /usr/local/bin/httpd
```

### Overview for the Impatient – PHP

#### # EXTRACT AND INSTALL

```
# su
# cd /home/your-user-name-here/
# tar xzf php-4.3.1.tar.gz
# ln -s php-4.3.1 php
# cd ./php
```

#### # CONFIGURE AND INSTALL

```
# ./configure \
--with-apxs2=/usr/local/apache/bin/apxs \
--with-mysql=/usr/local/mysql \
--prefix=/usr/local/apache/php \
--with-config-file-path=/usr/local/apache/php \
--enable-track-vars \
--enable-force-cgi-redirect \
--disable-cgi \
--with-zlib \
--with-gettext
```

```
# make
# make install
```

#### # COPY php.ini INTO PHP DIRECTORY

```
# cp -p php.ini-recommended /usr/local/apache/php/php.ini
```

#### # EDIT httpd.conf

```
# cd /usr/local/apache/conf/
```

Open http.conf with an editor. In the LoadModule section add the following line, or uncomment if it is already there.

```
LoadModule php4_module      modules/libphp4.so
```

It should appear once and only once. Your DirectoryIndex should look like this, add what is needed to mirror this line:

```
DirectoryIndex index.html index.php default.php
```

In the AddType section, add the following lines:

```
AddType application/x-httpd-php php
AddType application/x-httpd-php-source phps
```

I have read extensively on the install process and it seems the best approach is to install MySQL, then Apache, then configure for PHP.

### MySQL Installation

For further reference, refer to “Build Your Own Database Driven Website Using PHP & MySQL” by Kevin Yank or “MySQL Visual Quick Start Guide” by Larry Ullman as listed above in the references.

Before you begin the installation, create a user mysql with a password of your choice. You can do this from the command line with the following:

```
# useradd mysql
```

Or through the GUI through System Settings > Users and Groups. Be sure to create a mysql group for this user.

Open a new terminal. Become root and Cd into the directory where you downloaded the files. In my case it was /home/ajberry. I want to install it to /usr/local. Once extracted, I created a symbolic link in /usr/local. This way, I can upgrade versions and just change the symbolic link. Become root:

```
# su
Password:
```

Extract the Archive:

```
# tar xzf mysql-standard-4.0.12-pc-linux-i686.tar.gz
```

Next we need to create a symbolic link in /usr/local, so we can reference as just mysql

```
# cd /usr/local
# ln -s /home/your-user-name-here/mysql-standard-4.0.12-pc-linux-i686
mysql
# ls
bin  downloads  etc  games  include  lib  libexec  mysql  sbin  share
src
```

If mysql is baby blue, you have successfully made the symbolic link. Cd into the `/usr/local/mysql` directory. We now must install the database files. Do so with the following command.

```
# scripts/mysql_install_db
```

Now that MySQL is installed we need to firm up its security. By default, MySQL creates a root user with a blank password with complete ownership. We need to change this. If you followed the first instruction, you already have a user mysql in mysql group. If not, please refer to the first instruction above before going any further. We are going to change ownership of the data directory to user mysql.

```
# cd /usr/local/mysql
# chown -R mysql data
# chgrp -R mysql .
# chmod -R go-rwx data
```

Next, we will start the daemon for the first time. Then, request its status

```
# bin/safe_mysqld --user=mysql &

# bin/mysqladmin -u root status
```

You should see some stats, something like this:

```
[ajberry@localhost ajberry]$ mysqladmin -u root status
Uptime: 29525  Threads: 1  Questions: 16  Slow queries: 0  Opens: 7
Flush tables: 1  Open tables: 1  Queries per second avg: 0.001
```

If the error message says mysql daemon ended, look for a file in the `usr/local/mysql/data` directory named `hostname.err` (where `hostname` = the machine's host name).

Now that the server is running, we want to set it up to start at boot. The first step is creating a file called `my.cnf` in the `/etc` directory. This file will be looked at each boot, telling the OS that each time the `mysqld` daemon is loaded to use the user `mysql` that was created.

```
# cd /etc
# touch my.cnf
# vi my.cnf
```

NOTE THE FOLLOWING ARE VI COMMANDS SPECIFIC TO THE VI EDITOR, CONSULT YOUR EDITORS MAN PAGES TO FIND THE EQUIVALENT COMMANDS FOR YOUR EDITOR

Press **I** to insert then enter the following lines:

```
[mysqld]
user=mysql
```

Press ESC then type:

```
:wq
```

The file has been written and saved. Whenever the mysql daemon is loaded, it automatically starts as the mysql user. We need to shut down the server now and copy the mysql.server file into the startup directory.

```
# cd /usr/local/mysql
# bin/mysqladmin -u root shutdown
# chmod u+x support-files/mysql.server
# support-files/mysql.server start
# bin/mysqladmin -u root status

# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/
# cd /etc/init.d
# chmod 755 mysql.server
# cd /etc/rc2.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc3.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc5.d
# ln -s ../init.d/mysql.server S99mysql
# cd /etc/rc0.d
# ln -s ../init.d/mysql.server K01mysql
```

At this point, you should have mySQL installed and configured to boot at startup. To test, reboot the system then request the status of the server.

```
# shutdown -r now
# cd /usr/local/mysql
# bin/mysqladmin -u root status
```

If you see some stats like before, the daemon is running. MySQL installation is finished. You may have noticed that several commands referenced mysqladmin, and it would be wise to create a symbolic link in /usr/local/bin so you can reference it from anywhere.

```
# ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql
# ln -s /usr/local/mysql/bin/mysqladmin /usr/local/bin/mysqladmin
# ln -s /usr/local/mysql/bin/mysqldump /usr/local/bin/mysqldump
```

MySQL is now installed and configured. You may wish to check out a GUI interface for MySQL called mysqlcc. It is available from the MySQL downloads page.

### Apache 2.0.45 Installation

For further reference, refer to “Beginning PHP4”, “Professional PHP4”, and <http://dan.drydog.com/apache2php.html> as referenced above.

If you still have a terminal open, Cd into the directory where the files were downloaded. If not, simply open a new terminal and do.

Cd into the downloads directory:

```
# cd /home/your-user-name-here/
```

Become root:

```
# su
Password:
```

Extract the Archive:

```
# tar xzf httpd-2.0.45.tar.gz
```

After the file is extracted move into the source directory, or you may wish to create a symbolic link to lighten your typing load. To create a symbolic link, do the following:

```
# ln -s httpd-2.0.45 httpd
```

```
# cd ./httpd
```

We must now configure apache with several additional modules. In the source directory type the following:

```
# ./configure --prefix=/usr/local/apache \  
--enable-so \  
--enable-cgi \  
--enable-info \  
--enable-rewrite \  
--enable-speling \  
--enable-usertrack
```

Note that only the `--enable-so` is needed, the rest is specific to my configuration. For example, I enabled `ssl` for secure transactions. You could put that all on the same line, but in my experience, there are less errors this way. Make sure the last line does not have a slash at the end.

Description of enabled modules:

- `--enable-so` : Turns on DSO capability
- `--enable-cgi` : enable CGI scripts
- `--enable-info` : query server information
- `--enable-rewrite` : regex URL translation
- `--enable-speling` : correct common URL misspellings
- `--enable-usertrack` : user-session tracking

For a complete list of options, please see the Compiling and Installing section of the Apache 2 documentation. Typing `./configure --help` will also provide some assistance if you wish to configure more options.

If no errors were thrown proceed with a make.

```
# make
```

Then

```
# make install
```

To test your installation, cd into the Apache directory and type the following:

```
# cd /usr/local/apache
#bin/apachectl start
```

Apache should be up and running. Launch a browser and point it to <http://localhost>. The Apache test page should pop up.

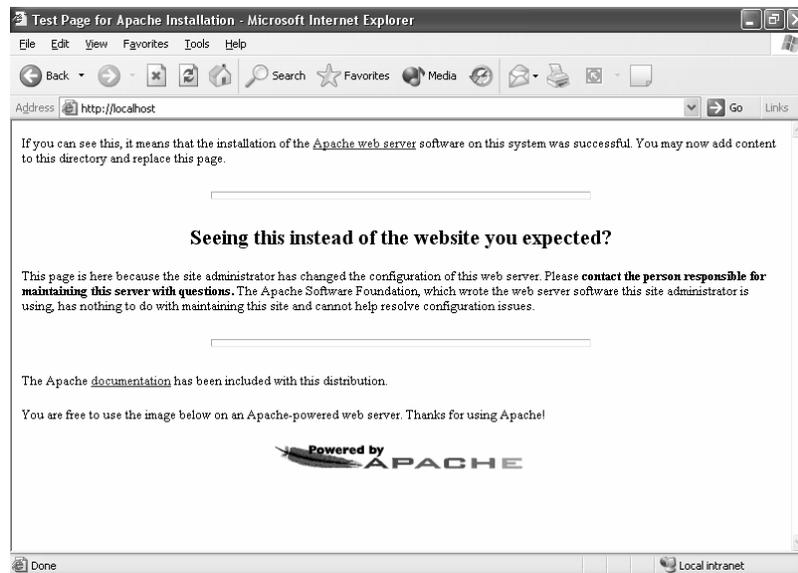


Figure 3.6

We now want to configure the Apache daemon to start automatically at startup as we did with mysql. We do this by copying the apachectl file into the `/etc/init.d/` directory. Notice that the file was renamed `httpd`.

```
# cp /usr/local/apache/bin/apachectl /etc/init.d/httpd
# cd /etc/init.d
# chmod 755 httpd
# cd /etc/rc2.d
# ln -s ../init.d/httpd S99httpd
# cd /etc/rc3.d
# ln -s ../init.d/httpd S99httpd
```

```
# cd /etc/rc5.d
# ln -s ../init.d/httpd S99httpd
# cd /etc/rc0.d
# ln -s ../init.d/httpd S99httpd
```

Apache is now configured to run at startup. We can now move on to configuring PHP, after we shutdown Apache and create a symbolic link so that we can start/stop the daemon from anywhere.

```
# cd /usr/local/apache/
# bin/apachectl stop

# ln -s /usr/local/apache/bin/apachectl /usr/local/bin/httpd
```

We can now start and stop the httpd daemon with a call from anywhere by simply typing:

```
# httpd start
```

or

```
# httpd stop
```

### PHP 4.3.1 Installation

For further reference, refer to “Beginning PHP4”, “Professional PHP4”, and <http://dan.drydog.com/apache2php.html> as referenced above.

If you still have a terminal open, Cd into the directory where the files were downloaded. If not, simply open a new terminal and do.

Cd into the downloads directory:

```
# cd /home/your-user-name-here/
```

Become root:

```
# su
Password:
```

Extract the Archive:

```
# tar xzf php-4.3.1.tar.gz
```

After the file is extracted move into the source directory, or you may wish to create a symbolic link to lighten your typing load. To create a symbolic link, do the following:

```
# ln -s php-4.3.1 php
# cd ./php
```

We must now configure apache with several additional modules. In the source directory type the following:

Don't forget the slashes at the end of each line except the last

```
# ./configure \
--with-apxs2=/usr/local/apache/bin/apxs \
--with-mysql=/usr/local/mysql \
--prefix=/usr/local/apache/php \
--with-config-file-path=/usr/local/apache/php \
--enable-track-vars \
--enable-force-cgi-redirect \
--disable-cgi \
--with-zlib \
--with-gettext
```

We have added several options such as mySQL support (make sure you specify full path to mysql), disabled cgi support, since we are adding PHP as a DSO(Dynamic System Object) in Apache. We have also changed the location of our `php.ini` file, and installed the command line version of PHP.

Description of enabled modules:

- `--with-apxs2[=file]` : Build shared Apache 2.x module. File is the optional pathname to the Apache apxs2 tool
- `--with-mysql[=DIR]` : Include MySQL support. DIR is the MySQL base directory. If unspecified, the bundled MySQL library will be used.
- `--prefix` : Sets the path to install PHP
- `--with-config-file-path` : Sets the path in which to look for `php.ini`, defaults to `PREFIX/lib`
- `--disable-cgi` : Disable building CGI version of PHP
- `--enable-force-cgi-redirect` : Enable the security check for internal server redirects.
- `--with-zlib` : Include ZLIB support (requires `zlib >= 1.0.9`)
- `--with-gettext` : Include GNU gettext support

Again, for additional configuration, type `./configure --help` or see the Installation chapter in the PHP Manual.

If no errors were thrown proceed with a make.

```
# make
```

Then

```
# make install
```

PHP should now be installed, but we are not done yet. We have to tweak a few files to get everything to work.

First, we need to install the `php.ini` file into the `/usr/local/apache/php/` directory.

```
# cp -p php.ini-recommended /usr/local/apache/php/php.ini
```

Next, we need to add a few things to Apache's `httpd.conf` file located in `/usr/local/apache/conf/`. First, we need to tell apache to load the `php4_module`. Second, we must add `index.php` as a default document, meaning if `index.php` exists in the top directory of a web directory, that page will load by default if none other is specified. We also need to specify what application will handle the interpreting of `php` files. Finally, we enable syntax coloring for debugging purposes.

```
# cd /usr/local/apache/conf/
```

Open the `httpd.conf` with an editor. Locate the section labeled `LoadModule` and add the following line.

```
LoadModule php4_module          modules/libphp4.so
```

It should appear once and only once. Next, locate the `DirectoryIndex` line and make sure it has as least the following:

```
DirectoryIndex index.html index.php default.php
```

The next section you are looking for is `AddType`. We must specify which application handles files with a `php` extension. We also want to enable color-coding for `php` code so it will be easier to debug. Add the following lines to that section.

```
AddType application/x-httpd-php php
AddType application/x-httpd-php-source phps
```

Next we need to make sure there is not an instance of `SetOutputFilter` installed. `AddType` and `SetOutputfilter` can be used interchangeably, but not together. Look for a line starting with `SetOutputFilter`, if you cannot find one, this is a good thing. If it is present, comment it out using leading `#`'s.

One other thing before we write the file that may prove useful is to know the root folder for web documents. You can find this by looking for the `DocumentRoot` Section. It is usually `/usr/local/apache/htdocs` by default, but you can change to reflect your tastes. Write down or remember what your document root directory was because you will need that in a minute. Write the file.

## Testing the Trio

First, start Apache,

```
# httpd start
```

Next open a text-editor and enter the following:

```
<html>
  <head><title>PHP Test</title></head>
  <body>
    <h2>PHP Information</h2>
    <p>
      <?php phpinfo(); ?>
    </p>
  </body>
</html>
```

Save the file as `phpinfo.php`. Launch a new browser and point it to `http://localhost/phpinfo.php`. You should see information about the install. It should look something like the following:

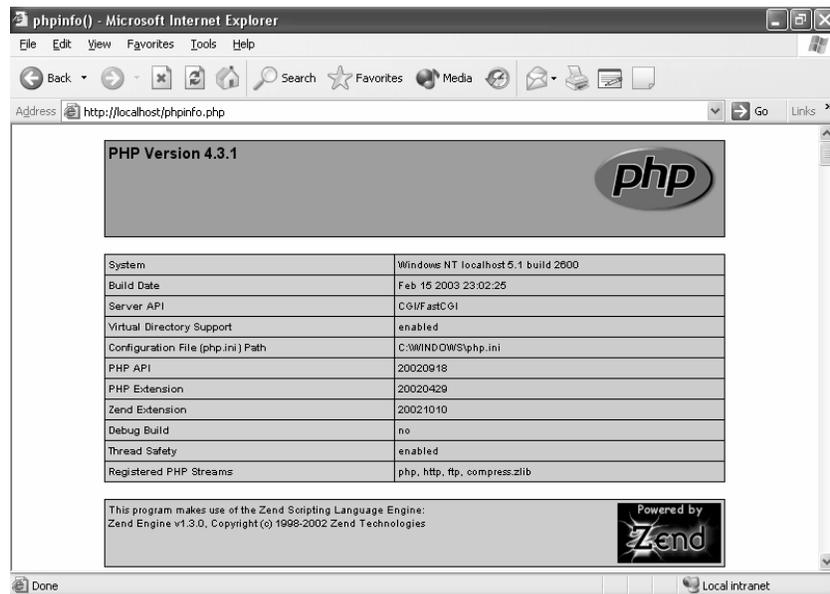


Figure 3.7

MySQL/Apache/PHP is up and running. Happy developing.

## Final Results/Recommendations

If using this document helps you to accomplish this task, please email at [ajberry12@earthlink.net](mailto:ajberry12@earthlink.net). I would be interested in your configuration and distribution. Eventually, I would like to compile a list of known good configurations that this guide pertains to.

I feel it only fair to mention that I successfully installed the trio in under 30 minutes on a Windows XP machine, with first time success. What fun was that you ask? Not much, but it does point out a major flaw in open source installations. Most windows installations will install the first time with little trouble, the need for documentation exists more in the user arena, not that of the installer. The inverse is true in the open source community. I will update as I see fit. Please forward questions and comments to [ajberry12@earthlink.net](mailto:ajberry12@earthlink.net).

### References

- Anderson, Dan. "Apache 2 and PHP (mod\_php) on Linux". 31 May, 2003. 6 June 2003  
<<http://dan.drydog.com/apache2php.html>>
- Apache 2.x/PHP/MySQL Installation. Linux Guruz. 2003. 6 June, 2003  
<<http://www.linuxguruz.com/z.php?id=322>>
- Ullman, Larry. *MySQL Visual Quickstart Guide*. 2003. ISBN 0-321-12731-5
- Wrox Press. *Beginning PHP4*, 2<sup>nd</sup> Edition. 2003. ISBN 1-861003-73-0
- Wrox Press. *Professional PHP4*, 2<sup>nd</sup> Edition. 2003 ISBN 1-861006-91-8
- Yank, Kevin. *Build Your Own Database Driven Website Using PHP & MySQL*, 2<sup>nd</sup> Edition. March 2003. ISBN 0-9579218-1-0

**Product:** Oscar

**Product Type:** Cluster

**Author:** Todd Franklin, Mohana Varmshi Gudepu, C. Judith Nyabando. Gunter Wambaugh

---

### **Problem Background**

Hari Machineni<sup>5</sup> wants to configure the Beowulf cluster in Wilson Wallis computer lab to use Oscar. Before installing Oscar on the Beowulf cluster he wanted it tested independently.

### **Product Placement**

Open Source Cluster Application Resource (OSCAR) is a cluster.

A cluster is a group of computers connected together on a network that work as a single machine. Each machine on a cluster is referred to as a node. Usually a single node will be the server and the rest are clients. The server node serves requests to clients and the clients perform the computations. OSCAR software package was designed to perform high performance computing and by default it provides common packages for high performance computing such as MPI, Parallel Virtual Machine (PVM), and Portable Batch System (PBS). The requirements for the project were to test the following:

- Installation of OSCAR
- Node Management: Addition and deletion of cluster nodes
- Workload management
- Remote package retrieval
- Switcher tool

The customer also wanted a detail report of what tools are installed, how they are installed and what tests have been performed to verify that they are installed.

### **Criteria**

According to the customer and Jeremy Enos the requirements listed above are met if the following tools are configured.

- C3 - Cluster Management Tools (ORNL)
- SIS - Network OS Installer (IBM)
- MPI-CH - Message Passing Interface
- LAM - Message Passing Interface (Indiana University)
- SSH- Secure Transactions
- PBS - Job Queuing System
- Maui - Batch scheduler
- PVM - Parallel Virtual Machine (ORNL)
- Firewall/NAT:
- Monitoring

Therefore our focus was directed to testing the above tool. We also installed Ganglia to meet the monitoring requirement.

---

<sup>5</sup> Hari Machineni is a graduate assistant responsible for taking care of the Beowulf cluster in Wilson Wallis lab

### Installation and Testing Overview

The installation process of OSCAR involves installing OSCAR and Linux packages on the server node, and building and installing client images on the client nodes. OSCAR can run on systems running Red Hat 7.1, 7.2, 7.3 and Mandrake 8.2, 9.0.

#### Prerequisites for the Server Node

- CPU of i586 or above
- A network interface card that supports a TCP/IP stack. 2 network cards are required if the server node will also be the router between the cluster nodes and an external network.
- At least 4GB total free space – 2GB under / and 2GB under /var
- An installed version of Linux (any of the distributions mentioned above)

#### Prerequisites for the client nodes

- CPU of i586 or above
- A disk on each client node, at least 2GB in size
- A network interface card that supports a TCP/IP stack
- Must be running the same Linux distribution and version as the server node
- All clients must have the same architecture
- Floppy or PXE enabled BIOS

### Installation

We installed Red Hat 7.2 on two machines. We downloaded an OSCAR distribution package (Oscar v2.2) from the OSCAR web site [OSCAR 2003] and unpacked it on the machine that would be the server. There are three flavors of the distribution: *regular*, *extra crispy*, and *secrete sauce*. We installed the regular flavor that has all the materials needed to run OSCAR. The hostname of the server node was changed to *oscar* and its IP address was set to 10.1.1.123. It is discouraged to use *localhost* as a hostname. The next step was to copy the RPMs from the Red Hat 7.2 CDs to `/tftpboot/rpm`. To run the OSCAR installer we moved into the *oscar* directory: `cd /root/oscar-2.2` and then executed the following command to setup and configure OSCAR:

```
# ./install cluster eth0
```

After the configuration was completed the installation wizard below was displayed.



Figure 3.8: Oscar Installation Wizard

The first 2 Steps are optional so we did not perform them.

Step 3: Install OSCAR Server packages - installs the packages for the server node.

SIS (System Installation Suite) was one of the tools that Hari Machineni wanted us to test. SIS is a network installer for Linux. SIS is used by OSCAR to build the client image and network install the image on the clients. This tool is also used to delete images. The installation steps involving SIS are discussed below.

Step 4: Build OSCAR client image - builds the image to be installed on the clients using SIS. We named the image oscarimage

Step 5: Define OSCAR client – we specified the base name for the nodes, oscarnode and specified the starting number to 1. Therefore the first client was named orscanode1. We set the IP address to 10.1.1.124 for the first client.

Step 6: Setup the network – this step is used to collect MAC addresses for the clients. In order to collect the MAC address we clicked the Start Collecting MACs button while the client was network booting. Network booting the client will install the image on the client. Before moving to step 7 we had to boot the client from the network and then from the hard drive again.

**Step 7: Complete Cluster Setup** - This performs the final installation configurations scripts, cleans up the cluster set up.

### Testing C3

C3 are a set of Cluster Management tools. The tools include cpush, cget and clist. Cpush pushes files to nodes, cget gets files from nodes and clist list clusters on a network. We tested clist:

```
clist
results: oscar direct local
```

### Testing SSH

By default OSCAR uses ssh instead of rsh for remote communication. Therefore when we ran Lamboot we did not have to set up cryptographic keys. However for testing purposes we ran the following commands from oscarn1 to oscar (the server) with the login name osuser: `ssh -l osuser oscar` and we were able to connect and run LAM.

### Testing Monitoring

To test the monitoring tools we had to install Ganglia. Once it was installed we executed the command `gstat` and got the following results:

```
CLUSTER INFORMATION
Name: unspecified
Hosts: 1
Gexec Hosts: 0
Dead Hosts: 0
Localtime: Tues Apr 15 17:58:43 2003
```

Note: Ganglia had to be installed before installing OSCAR.

### Testing LAM and MPI

LAM is a programming environment for MPI. LAM/MPI should only be run as an ordinary user and not as root. To test LAM and MPI we logged on as *osuser* and we created a file called *myhost* that had the names of the hosts on the cluster that will be used by a program. The contents of the file were:

```
oscar (the host name of the server)
oscarn1 (the host name of the client)
```

Then we run the command `Lamboot -v myhost` to start the LAM environment and the following lines were displayed:

```
LAM 6.5.9 - University of Indiana
Executing hboot on n0 (oscar.localhost - 1 CPU)...
Executing hboot on n1 (oscarn1.localhost - 1 CPU)...
```

Our test programs were `master.c` and `slave.c`. We executed the following commands:

```
mpicc master.c -o master
mpicc slave.c -o slave
mpirun myapp.in
```

`mpicc` is a MPI wrapper compiler for C++. To exit from the LAM environment we executed the `Lamhalt` command.

### **Lessons Learned/Problems**

The installation process of Oscar is simple and straight forward. However we ran into problems when we ran the `mpirun` command. The `mpirun myapp.in` command produced a file not found error. The master executable was not found. To determine the problem we seeded bugs in `master.c` and tried to compile it again that is when we discovered that the programs were not compiling at all. We could not compile and run C++ programs on the cluster using `mpicc`. We realized later that `mpicc` is an MPI wrapper compiler for C++ and not a C++ compiler. Therefore we needed to install `gcc` but due to time constraints we were not able to install `gcc`.

### **Final Results/Recommendations**

The tests we ran were successful except for running the test programs. However we concluded that the problem was not related to OSCAR or MPI but that we needed a C++ compiler. The other tests were implemented by performing step 8: Test Cluster Setup. This tested PBS, maui, PVM (via PBS), MPICH (via PBS), LAM/MPI (via PBS) and it will ssh server to client and client to server. All these tests passed.

### **References**

Enos, Jeremy. "Oscar". NCSA Cluster Group. 30 June 2002. 30 April 2003

<<http://oscar.sourceforge.net/talks/oscar-mit.ppt>>

OSCAR-Open Source Cluster Application Resources. 18 January 2003. 30 April 2003

<<http://oscar.sourceforge.net>>

**Product:** Shavlik HFNetChkLT

**Product Type:** Patch Tracking and Installation Software

**Author:** C. Jutjih Nyabando

---

### **Problem Background**

The project was done on behalf of Robert Nielsen, CSCI Systems Manager. There is the need to keep up to date with patches for the software on computers in the Computer Science department. It is difficult to manually check for missing patches and then install them. Automating this process would save time and reduce errors.

### **Product Placement**

HFNetChkLT is a free patch tracking software from Shavlik that checks and install missing patches on a machine. The tracking and installation can be done on a single machine, a group of machines or on all machines in a domain at any given time. The advantage of using a patch tracking tool is that it performs automatic checks and installation for missing patches on a particular machine. This reduces the amount of time required to do it manually. It also reduces human errors.

### **Criteria**

The purpose of the project was to test the effectiveness of HFNetChkLT in tracking and installing missing patches. The customer's intention is to use this software to trace and install patches on faculty machines, therefore test tracking and installation tests were run on Steven Jenkins'<sup>6</sup> desktop. Tracking tests were also run remotely on one of the computers in the computer lab but installation tests were not performed on the remote machine because I did not acquire the permission to do so. These test had to be performed on faculty machines.

### **Installation Overview**

HFNetChkLT can be downloaded from Shavlik's web site for free. However you must register to get it for free and it is only free if it is to be used on fifty or less machines. The software can be installed on machines running Windows NT 4.0, Windows 2000, and Windows XP and the machines must have:

- Internet Explorer 5.5 or later
- Microsoft Data Access Components (MDAC) 2.6 SP2 or later
- Microsoft Windows Installer version 2.0
- Microsoft XML Parser 3.0 SP2 or later
- Microsoft Jet 4.0 SP3 or later

If any of these requirements are missing the installer will try to download and install them or it will give the URL to the websites where the requirements can be obtained. Administrative access is required to install and use the software. To install the software one need to just double click the set up icon and follow instructions on each dialog.

---

<sup>6</sup> Mr. Steven Jenkins is the CSCI5360 instructor. He volunteered his computer for test purposes.

## Product Analysis

Overall, the HFNetChkLT is a good product. It is easy to install and easy to use. To perform a scan on the local machine just click on the *machine* option under *Scan What* option and then select begin scan. The first scan that was performed on the local machine took about three minutes and subsequent scans took less than a minute. The installation process for patches is also simple. It took about 15 minutes to download and install 13 out of 17 missing patches. The other 4 were not downloaded and installed. This does not include the time it took the machine to reboot. The results for each scan and patch deployment are saved and detailed reports are also available.

## Final Results/Recommendations

Figure 1 and 2 below shows a screen shot of the results of the scan on JENKINS, Steven Jenkins' computer. 17 patches were missing on JENKINS. The results, as shown in figure 1 list the type of patch, item, software product that needs the patch and a brief description of the patch. Scanning tests were run on a remote machine, WINXP00 and the scan indicated that 13 patches were installed, and the Office XP SP1 service pack and 3 patches were missing (see figure 3.11).

The screenshot displays the Shavlik HFNetChkLT (Unregistered) interface. The main window shows a table of scan results for machine JENKINS. The table has columns for Type, Item, QNumber, Deployment, Product, Description, and Comment. The results are as follows:

Type	Item	QNumber	Deployment	Product	Description	Comment
Missing Patch	MS02-050	Q329115		Windows XP Pr...	Certificate Valid...	
Missing Patch	MS02-054	Q329048		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS02-055	Q323255		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS02-059	Q330008		Excel 2002 SP2	Flaw in Word Fi...	
Missing Patch	MS02-059	Q330008		Word 2002 SP2	Flaw in Word Fi...	
Missing Patch	MS02-063	Q329834		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS02-067	Q331866		Outlook 2002 S...	E-mail Header ...	
Missing Patch	MS02-070	Q329170		Windows XP Pr...	Flaw in SMB Si...	
Missing Patch	MS02-071	Q328310		Windows XP Pr...	Flaw in Window...	
Missing Patch	MS02-072	Q329390		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS03-001	Q810833		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS03-003	Q812262		Outlook 2002 S...	Flaw in How Ou...	
Missing Patch	MS03-004	Q810847		Internet Explore...	Cumulative Pat...	
Missing Patch	MS03-005	Q810577		Windows XP Pr...	Unchecked Buf...	
Missing Patch	MS03-008	Q814078		Windows XP Pr...	Flaw in Window...	
Missing Patch	MS03-010	Q331953		Windows XP Pr...	Flaw in RPC En...	
Missing Patch	MS03-011	Q816093		Windows XP Pr...	Flaw in Microso...	
Informational Item	NET Fra...	SP2		.NET Framework...	.NET Framework...	
Informational Item	Access 2...	SP2		Access 2002 S...	Access 2002 S...	
Informational Item	MDAC 2.7	SP1		MDAC 2.7 SP1	MDAC 2.7 SP1	
Informational Item	Office XP	SP2		Office XP SP2	Office XP SP2	
Informational Item	PowerPoi...	SP2		PowerPoint 200...	PowerPoint 200...	
Informational Item	Windows ...	SP1		Windows Medi...	Windows Medi...	

The 'Patch Details' window is open for 'Windows Media Player for Windows XP Windows Media Player for Windows XP SP1'. It shows the status as 'Informational Item' and has tabs for Patch Info, TruSecure, Missing, and Installed. The status bar at the bottom indicates the scan was performed on 4/14/2003 at 2:16:03 PM and that the user is running HFNetChkPro version 4.0.73.

Figure 3.9: Detailed patch scan report

Installed Products	✓ Patches Found	✗ Missing Patches	✗ Missing Service Packs
.NET Framework SP2	0	0	0
Access 2002 SP2	0	0	0
Excel 2002 SP2	0	1	0
Internet Explorer 6 SP1	0	1	0
MDAC 2.7 SP1	0	0	0
Office XP SP2	0	0	0
Outlook 2002 SP2	0	2	0
PowerPoint 2002 SP2	0	0	0
Windows Media Player for	0	0	0
Windows XP SP1	0	0	0
Windows XP Professional SP1	0	12	0
Word 2002 SP2	0	1	0

Figure 3.10: Patch scan report by product

### Installation Results

There are three options to deploy patches: *install immediately*, *copy patche(s) to machine but do not install*, and *schedule installation time*. For this test *install immediately* was chosen. The table below shows the results of the scan on JENKINS after second installation of the patches. The first time 13 out of 17 missing patches were downloaded and installed. Another scan was performed and it showed that 13 patches were found and 4 were missing. Patch deployment was performed again and 2 out of 4 patches were installed. The other 2 were not downloaded and installed. After the patches are installed the machine should reboot.

<u>Windows XP Professional SP1 MS02-050, Q329115</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:29 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-054, Q329048</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:29 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-055, Q323255</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:29 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Word 2002 SP2 MS02-059, Q330008</u>	Installation Failed	Patch copy complete at 4/16/2003 10:00:30 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Excel 2002 SP2 MS02-059, Q330008</u>	Executed - Pending Reboot	Patch copy complete at 4/16/2003 10:00:30 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-063, Q329834</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:30 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-070, Q329170</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:30 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-071, Q328310</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:30 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS02-072, Q329390</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS03-001, Q810833</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Outlook 2002 SP2 MS03-003, Q812262</u>	Installation Failed	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Internet Explorer 6 SP1 MS03-004, Q810847</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS03-005, Q810577</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS03-008, Q814078</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:31 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS03-010, Q331953</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:32 AM Scheduling successful at 4/16/2003 10:00:33 AM
<u>Windows XP Professional SP1 MS03-011, Q816093</u>	Installation Succeeded	Patch copy complete at 4/16/2003 10:00:32 AM Scheduling successful at 4/16/2003 10:00:33 AM

**Missing Service Packs = 1**  
**Patches Missing = 3**  
**Patches Found = 13**

Installed Products	✔ Patches Found	✘ Missing Patches	⚠ Missing Service Packs
.NET Framework SP2	0	0	0
Access 2002 SP1	0	0	0
Excel 2002 SP1	0	0	0
Internet Explorer 6 SP1	1	0	0
MDAC 2.7 SP1	0	0	0
Office XP SP1	0	0	1
Outlook 2002 SP1	0	1	0
PowerPoint 2002 SP1	0	0	0
Windows Media Player for	0	0	0
Windows XP SP1			
Windows XP Professional SP1	12	2	0
Word 2002 SP1	0	0	0

Figure 3.11: Patch scan report for the remote machine WINXP00

### Recommendations

The results from the tests indicate that HFNetChkLT is effective. It was able to download and install 77% of the missing patches the first time within a reasonable time frame.

### References

Shavlik Technologies, LLC. 30 April 2003 <<http://www.shavlik.com>>

**Product:** Sun One Active Server Pages

**Product Type:** Server-Side Web Development Application

**Author:** Adam Berry

---

### **Problem Background**

The problem with ASP is that it has historically been a Microsoft only product. To use ASP, a developer had to be running a flavor of Internet Information Services, Microsoft's Web Server. The problem with IIS is its constant security bulletins and patches. Now, there are other alternatives, one being Sun One Active Server Pages.

### **Product Placement**

Active Server Pages is a Microsoft technology solution for server-side web development. When an html page is requested by the client, the server interprets the ASP code and fills in the relevant data based on the code. ASP supports connections from databases, files, etc, making it a powerful tool. Many E-commerce sites use ASP for the dynamic database driven environment.

Sun One Active Server Pages is an application by Sun Microsystems. It has also been formerly known as ChiliSoft! ASP. The application allows the porting of ASP to many UNIX based environments, including Apache. Its native environment would be Sun One Web Server running on Solaris.

### **Installation Overview**

Test System

Software (First Round):

- Red Hat Linux 8.0 (2.4-18 kernel)
- Apache Web Server 1.13.19 (bundled with install)
- Sun One Active Server Pages 3.6.2

Software (Second Round):

- Red Hat Linux 7.3
- Apache Web Server 1.13.19 (bundled with install)
- Sun One Active Server Pages 3.6.2

Hardware:

- Tyan 2507D Tiger 230 Motherboard
- (2) PIII 1.13GHZ
- 784MB PC133 RAM
- 40GB WD HDD
- ATI Radeon 7000VE 32MB
- HP 8100i 4X CDRW

While Sun One Active Server Pages is \$495 per server license, they do offer a fully functional developers edition. To publish, the application must be purchased. They also offer a 30-day trial download for anybody wishing to try out the software. I downloaded the developer's edition.

Tar File:

<http://download.chilisoft.com/chiliasp/linux/chiliasp-3.6.2L.1047a.tar>

Iso Image:

<http://download.chilisoft.com/chiliasp/linux/chiliasp-3.6.2L.1047a.iso>

I tried both the tar and iso versions. I burnt the ISO using Nero Burning Rom. There was little difficulty with either method. Speed was also not really an issue.

CD-ROM Install (as root):

```
# mkdir mount
# mkdir cdrom
```

MOUNT CD

```
# mount cdrom -t iso9660 /dev/cdrom /mount/cdrom
```

OPEN TERMINAL

```
# cd /mount/cdrom
# ./install.sh
```

FOLLOW PROMPTS

Tar Install (as root):

```
# tar -xvf casp-3.6.2.linux.tar
# ./install.sh
```

The install process creates the necessary directories for you. A nice feature that it had was a certified bundled version of Apache Web Server within the install. I chose to install the bundled version, even though I already had Apache on the machine. This way I could choose default configurations without messing with my existing working server.

I chose to have all options run at startup and finished the install. An install summary is created for you by default. After viewing the summary and finding no errors, it was time to test.

After rebooting, I tried to access the sample content. It acted for a second as if it was processing, then threw an error:

```
Error 501: ASP Service Disabled
```

At first I thought that maybe all services hadn't been brought up at startup. I checked the logs by typing:

```
# opt/casp/asp-server-3000/caspctrl viewlog
```

I found that all services were said to be running. I then stopped the service with:

```
# opt/casp/asp-server-3000/caspctrl stopall
```

and tried to restart with:

```
# opt/casp/asp-server-3000/caspctrl startall
```

but to no avail. I still got the same error. After reading some documentation, which is actually quite easy to understand, I came up empty. I uninstalled everything and started from scratch.

With a fresh Linux Install, with almost no packages except the essentials, I tried again, this time using the Tar install. Same result, more wasted time. At this point, I was very frustrated. I was still getting:

```
Error 501: ASP Service Disabled
```

My next course of action was the Knowledge Base and Support Forums at sun.com. I browsed through some forum threads and found someone with a similar problem running the same setup (Red Hat 8, bundled Apache, Sun One 3.6.2). An administrator referred him to the following knowledgebase entry:

Article ID	Product	Category	Platform	Updated
200210161	ASP	General	Linux	4/13/2003

**Problem:**

Sun ONE Active Server Pages 3.6.2 for Linux doesn't work with RedHat 8.0, why?

**Solution:**

This is due to glibc incompatibilities in the newer distributions. From our documentation (see QuickStart in the download package), we support 2.4 kernels and up to glibc 2.2.4. The glibc problems may also extend to other Linux distributions that use newer system libraries. At first thought, I pondered simply downgrading my glibc to 2.2.4. At this point, I had spent quite a bit of time with setup and didn't want to take any chances. It was recommended that I install Red Hat 7.3 and go from there.

With a fresh install of Red Hat 7.3, I once again ran the installer program, which told me installation was successful. After firing up apache to <http://localhost/>, I was greeted with the Apache start page. Next, I tried the sample content at <http://localhost/caspsamp>, which threw the exact same error:

```
Error 501: ASP Service Disabled
```

After searching more threads in the support forums, it was discovered that the ASP engine was dying on any and all requests. I checked the appropriate logs to find similar circumstances but couldn't quite match anything up. The advice from an administrator was to uninstall and reinstall, not what I wanted to hear.

After running the `uninstall.sh` script provided, I reinstalled for the fourth time. I accepted defaults on everything, thinking that would be my best option for a successful install. Low and behold, when I fired up mozilla and went to <http://localhost/caspsamp>, there was the sample content. I was dumbfounded, yet excited. I toured the samples, which I thought were elementary, they didn't actually show you the capabilities of the software, they walked you through how to do some basic stuff in ASP.

I copied an ASP site I had designed last year into my DocumentRoot directory and edited my `httpd.conf` file to include `index.asp` in the `DirectoryIndex`. When I tried accessing my ASP site, I got the following error:

```
ADODB.Connection.1error '800a0bb9'
```

```
The application is using arguments that are of the wrong type, are out of acceptable range, or are in conflict with one another.
```

```
/king/DB/index.asp, line 29
```

It didn't take long to figure out that Microsoft Access databases will only run on a Windows platform. While Access is not considered a production level database platform, it is useful for small scale websites for which my existing code was built for. The following knowledge base article pointed me in the right direction.

Article ID	Product	Category	Platform	Updated
200110123	Sun Chili!Soft ASP	Database (ADO)	MS-Windows	4/15/2003

**Problem:**

How do I connect to Microsoft databases?

**Solution:**

You need a Windows-OS machine to access Microsoft databases. This is not a Sun Chili!Soft ASP requirement; no Microsoft application runs on Linux, and databases are no exception.

For Microsoft SQL Server 7.0, there is a direct driver. For SQL Server 6.5 and Access, connectivity is available through the use of DataDirect's (formerly MERANT's) SequeLink product. This is an ODBC driver in two parts; the UNIX portion ships as part of the Sun Chili!Soft ASP installation, and the Windows portion is available from our FTP site at <ftp://ftp.chilisoft.com/pub>. Log in anonymously and download the `slkntsrv.zip` file.

After downloading the sequelink tool, I installed it on a Windows XP machine. I accepted defaults for the installation and shared the directory that housed my Access Database. I then switched back to my Linux box and ran the setsqlnk utility from a terminal to create a DSN:

```
# cd /opt/casp/asp-server-3000/
# ./setsqlnk
```

```
[root@raq3 asp-apache-3000]# ./setsqlnk
SequeLink Connect Administration Tool on Linux
(c)Copyright 1995-1998 INTERSOLU, Inc., All rights reserved

The following Data Source is selected : .
[1] Select a Data Source
[2] New
[7] About
[0] Cancel
Select an action [0]:
```

Figure 3.12

See #2 in the How-to section of the References for a step-by-step walkthrough.

After my DSN was created, I used the built in connection testing tools to see if the DSN was created successfully. Low and behold, it passed the test on the first go-around.

However, I was not done yet. I must now setup a DSN through the Administration Console provided by Sun One Active Server Pages. The easiest way to get to the Admin Console is to go to the samples page at <http://localhost/caspsamp/> and click on the Admin Console link.

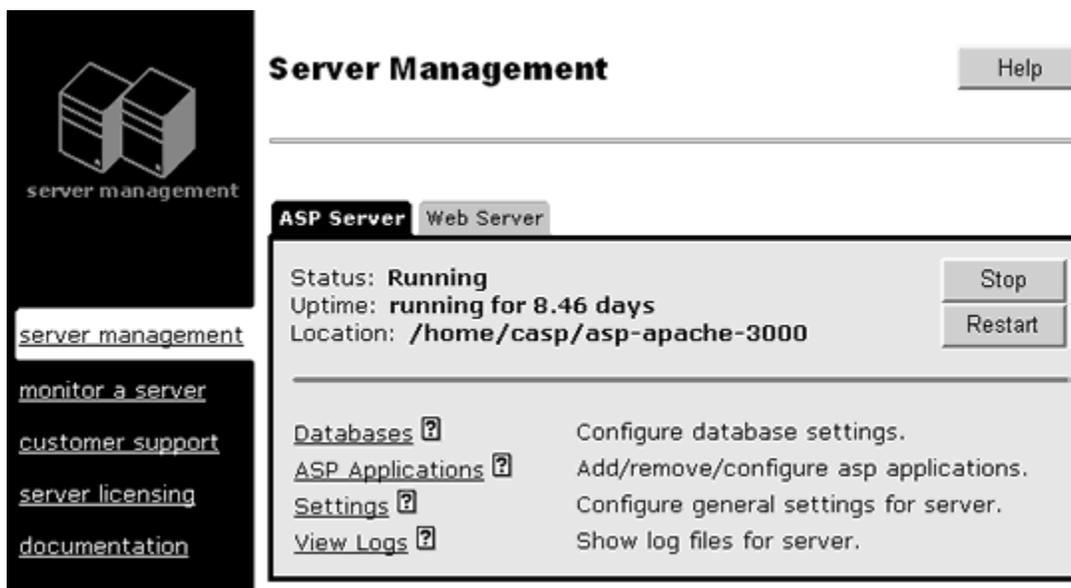


Figure 3.13: Server Management

Click Databases.

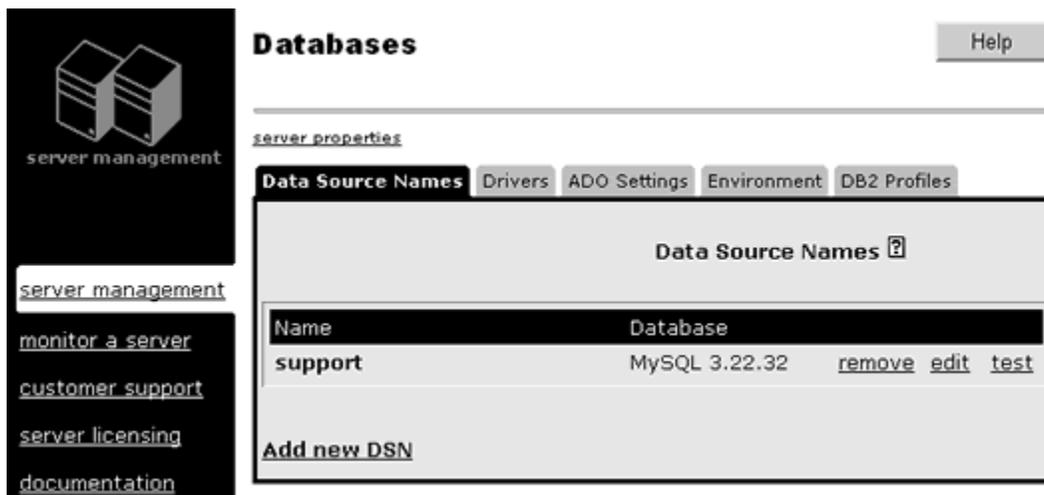


Figure 3.14: Database Management

Click Add New DSN, and fill in the appropriate fields.

The screenshot shows the 'New Data Source Name' configuration dialog. It has a 'Help' button and a 'server properties' section with tabs for 'Data Source Names', 'Drivers', 'ADO Settings', 'Environment', and 'DB2 Profiles'. The 'Data Source Names' tab is active. The dialog contains the following fields and options:

- DSN**: Text input field containing 'TestDSN'.
- Description**: Text input field containing 'This is a Test'.
- Database type**: Dropdown menu showing 'SequeLink 4.51a' with a 'select' button.
- Driver**: Text input field containing 'YYslk13.so V. 1.3'. Below it is a **NOTE** stating: 'You must use *'setsqlnk'* tool located in *'/home/casp/asp-apache-3000'* order to create an ODBC DSN using SequeLink.'
- SQLnkDSN**: Text input field.
- Database**: Text input field containing 'my\_database'.
- LogonID**: Text input field.
- Password**: Text input field.
- Confirm Password**: Text input field.

At the bottom are 'Save' and 'Cancel' buttons.

Figure 3.15: Database Configuration

After a successful test, I had to edit my asp code a little bit. I was using a DSN-less connection, which works flawlessly on a Windows machine, but refuses to work in Linux. This alone is a major drawback, especially if you do not have the ability to create your own DSN's on the hosting server. The good news was I only needed to change one line of code to make it work. A database connection is needed on virtually every page in the site I developed, so I simplified things by making one file (`config.asp`) and including it on every page. The following is my code example:

```
<html>
  <head>
    <title>CONFIGURATION</title>
    <link rel="stylesheet" type="text/css"
href="images/style/css/athletics.css">
  </head>
  <body>

  <%
Dim strcon
Dim objRec
Dim objCon

  `ORIGINAL LINE - MUST COMMENT OUT

  `strCon = "Driver={Microsoft Access Driver (*.mdb)}; DBQ=" &
  `Server.MapPath("db/king_athletics.mdb") &";"

  `NEW LINE

  strCon = "DSN=king"

  %>

  </body>
</html>
```

While I was optimistic that it was close to working, I took a chance and fired up Mozilla. I was delighted to see a successful loading of the `index.asp` page. I was curious if the Admin section would work, so that is the first place I headed. To my delight, I was able to add records to my database.

### Lessons Learned/Problems

The alternative to this madness is to run IIS from Microsoft. It is bundled with Windows 2000 and is a free download from their website. A major concern is the security of the system. However, for development purposes, you cannot beat its simplicity. It is very reliable and easy to configure. A novice can have IIS serving ASP pages in mere minutes after the install. In a nutshell, if you plan on using a Microsoft product, use their servers.

While anyone can appreciate a success, my opinion of the product has changed dramatically throughout the process. One negative fact is that, from start to finish, it took over 30 hours, 3 fresh installations, and various other resources to get working. For a developer, this is a real

hindrance. It also brings up a point about failed installations. The installs were made with the same cd's and same options chosen. In my case, 1 out of 4 installations was successful.

Although I did not like the fact that I had to use two machines to get a fully-functional ASP implementation, I am impressed with the product's performance. Access is not a production level database platform, and should not be held against the product because it does not natively support it. The bottom line is, Sun One Active Server pages were designed for use with production level database platforms like Oracle or MySQL. These DB platforms are natively supported and can be housed on one machine. I should also point out that a new release of Sun One Active Server Pages has been released since I first attempted this project. A major benefit to the 4.0 version is a built-in Microsoft Access to MySQL converter. I look forward to playing with that in my spare time.

### **Final Results/Recommendations**

ASP is best run on a Microsoft IIS server. However, I would not recommend hosting your own server running IIS. I would recommend using IIS for development purposes locally, and farm out the actual hosting to professional whose job is to provide security in web hosting.

There is hope for ASP in the open-source community, and with each new release, Sun One has shown improvement. With the new Access to MySQL converter, I believe more and more entry level developers won't be afraid to make the jump.

If you must host your own server, try a UNIX environment. If you need dynamic database driven websites, utilize technologies built for a UNIX platform like PHP. There are other technologies for web languages like CGI, Python and Perl.

### **References:**

- Community Forums. Sun Microsystems. 2003. 6 June, 2003 <<http://developer.sun.com/prodtech/webdir/community/forums>>
- Sun ONE Active Server Pages. Sun Microsystems. 2003. 6 June, 2003 <<http://developer.chilisoft.com/howto/createdsn.asp>>

**Product:** Virtual Network Computing (VNC) for Linux

**Product Type:** Console Emulation Utility

**Author:** Trey Buck

---

### **Problem Background**

The purpose of this project was to determine how well VNC (Virtual Network Computing) would suit the purposes of the Computer Science department at East Tennessee State University. The suitability of using VNC to gain remote access to a number of hosts in the Wilson-Wallis lab was the situation in question. The Wilson-Wallis lab houses a number of PCs running Red Hat Linux. What follows is a description of the installation and configuration of VNC version 3.3.3 on a PC running the Red Hat Linux distribution version 8.0. Both versions of VNC and Red Hat were the most current at the time of this writing.

### **Product Placement**

VNC is a software application that provides console emulation Servers and viewers have been developed for a variety of platforms such as DEC Alpha, PowerPC, Solaris, and most versions of Windows. Originally developed at what was most recently AT&T Laboratories Cambridge, it is currently available for use under the GNU Public License, making it a cost effective solution.

Systems running XWindows have long had the capability to access remote desktop sessions. This feature is an integral part of the XWindows architecture. The main advantages of using VNC for remote access are cost and consistency. VNC is a non-commercial package, freely available for use in any organization. VNC also provides a consistent remote access solution when used in multi-platform environments. VNC servers and clients are available for most major operating systems and can provide a unified remote access architecture, which can reduce training and maintenance costs.

### **Installation Overview**

Though Red Hat provides an RPM package for VNC, not all Linux distributions support RPM packages. The RPM was not used during this project, both to accurately gauge the complexity of the installation and provide documentation for other distributions. The precompiled binary files available on the VNC web site were used instead.

Installing the files is a simple matter of placing the appropriate files in the `/usr/bin/` directory. There are three files necessary to run the VNC server on Linux: `Xvnc`, `vncserver`, and `vncpasswd`. `Xvnc` is the actual server which provides the sessions. `Xvnc` can be started manually, but it is recommended that it be launched using the `vncserver` script. `vncserver` is a perl script which helps insulate the user from the myriad of option available for `Xvnc`. The third file, `vncpasswd`, is used to change the password for the sessions. It is not required that these files reside in the `/usr/bin/` directory. They may be placed anywhere, but it is useful to have them in a directory included in the path.

```
Xvnc          : VNC server daemon
vncserver     : script to launch Xvnc
vncpasswd     : utility to change the VNC session password
```

At this point the server is ready to run. To start the service, simply run the `vncserver` script. The script will prompt for the user for the password used to validate VNC sessions and ask for a confirmation. It will then start a new instance of the service and display a message similar to the following:

```
New 'X' desktop is localhost.localdomain:1

Starting applications specified in //.vnc/xstartup
Log file is //.vnc/localhost.localdomain:1.log
```

The most important item to notice is the last number on the first line, in this case '1'. This is the number of the display that the server is providing. The number of the display must be specified when connecting to the machine using a VNC viewer. The next line indicates where the `xstartup` file is located. This file is a script that determines what commands are run when the session begins. The window manager for the session is specified here. The last line gives the location of the log file. The `.vnc` folder and its contents are created by the `vncserver` script if they are not already present.

### Lessons Learned/Problems

Even though the server was ready to run, it was not very usable. The first issue that needed to be addressed was the window manager. VNC uses `twm` as its default window manager which provides somewhat less functionality than we required. The solution was to edit the `xstartup` file to specify a different window manager. GNOME was the preferred choice so the line "`twm &`" was replaced with the line "`gnome-session &`".

That change only brought us halfway to a better window manager. After making this change, GNOME failed to appear in the session window when the VNC client connected. A lengthy newsgroup search revealed that the `SESSION_MANAGER` environment variable needed to be unset to allow for multiple GNOME sessions to be active at once. Multiple sessions are necessary due to the fact that GNOME is often already running on the console. After this change was implemented in `xstartup`, the VNC session started with GNOME as expected. The `xstartup` file used is listed below.

```
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey

# don't start xterm or twm
#xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#twm &

# allow multiple gnome session and begin a new session
unset SESSION_MANAGER
gnome-session &
```

VNC was functional but there was room for improvement. In order to connect to the Linux host using VNC, it was necessary to log in and start the service manually. The user must also be logged on during a VNC session or the server will not know what settings to use for XWindows. Both of these problems needed to be solved. Once again a lengthy search on the web revealed some clues. The eventual solution was to include `vncserver` in the `rc.local` file. This allowed the VNC service to start when the system booted, not when a user logged on. After placing the appropriate reference in the `rc.local` file and rebooting the system, a message is displayed asking for the password used for validation. Entering and confirming the password set the system up correctly and allowed users to connect to the host using VNC after a reboot and before any user was logged on to the console.

### **Final Results/Recommendations**

The system was quite useable as configured, providing most of the functionality found at the console. There are some items that could be improved or reworked. Some tools, such as Users and Groups, are not functional and must be run from the console. Another issue is that, by starting the VNC service at boot time, VNC sessions will always be validated as root. This is a potential security hole. A method to begin a generic VNC session and prompt the user for their Linux credentials would be an excellent feature. To my knowledge there is no such mechanism for VNC.

### **References**

Virtual Network Computing. AT&T Laboratories Cambridge. 1999. 29 April 2003  
<<http://www.uk.research.att.com/vnc/>>

## VNC Installation Instructions for RedHat Linux

1. Obtain the VNC files. The homepage is: <http://www.uk.research.att.com/vnc/>
2. UnZip the files to a temporary directory.
3. Copy the following files to `/usr/bin/`:

```
Xvnc
vncserver
vncpasswd
```

4. Place the following line in `/etc/rc.local`:

```
vncserver
```

5. Reboot the system.
6. The system will display the following line:

```
You will require a password to access your desktops.
```

*Note: A password prompt will not display here but it wants a password and a confirmation.*

7. Type the password and press `<enter>`.
8. Retype the password and press `<enter>`.
9. Modify the `/.vnc/xstartup` file to start GNOME as follows:

```
#!/bin/sh

xrdb $HOME/.Xresources
xsetroot -solid grey

# don't start xterm or twm
#xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#twm &

# allow multiple gnome session and begin a new session
unset SESSION_MANAGER
gnome-session &
```

10. Test the service by connecting to the host from a VNC client (don't forget to specify which display to use. In most cases it's '1').

# *Utilities*

**Product:** RPM for PLT  
**Product Type:** Software Development  
**Author:** Gunter Wambaugh

---

### Problem Background

There was not an RPM for the latest version (203) of PLT.

### Product Goals

RPM is a package manager from Red Hat. It enables the distribution of pre-compiled software. RPM eliminates the hassle of having to compile software before installation. The following is an RPM spec file for PLT as well as a script used to automate the build process.

### Source Code and Documentation

The following is the RPM spec file I created for PLT:

```
Summary: Programming Environment
Name: plt
Version: 203
Release: 1
Source0: plt.src.x.tar.gz
License: LGPL
Group: Development/Languages
Vendor: Gunter Wambaugh <techgunter@yahoo.com>
URL: http://www.drscheme.org/
BuildRoot: %{_tmppath}/plt-root

%description
DrScheme, a pedagogical programming environment.

%prep

%setup -n plt/src

%build
CFLAGS="$RPM_OPT_FLAGS" ./configure \
    --prefix=%{_prefix} \
    --disable-debug

make

%install
rm -rf $RPM_BUILD_ROOT
mkdir -p %{_tmppath}/plt-root/usr
make prefix=$RPM_BUILD_ROOT%{_prefix} \
    install

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root)

%{_bindir}/*
```

```

%{_prefix}/collects/*
%{_prefix}/include/*
%{_prefix}/lib/*
%{_prefix}/man/*
%{_prefix}/notes/*
%{_prefix}/install

%post
cd %{_prefix}
./install
setup-plt

%changelog
* Tue Apr 22 2003 Gunter Wambaugh <techgunter@yahoo.com>
- Run the plt install script after plt is installed.
* Sat Mar 01 2003 Gunter Wambaugh <techgunter@yahoo.com>
- Initial build.

```

The following is the shell script I created to automate the build process:

```

#!/bin/sh
#
# [ 03.01.03 | Gunter Wambaugh ]
# Script for building a binary and source rpm.

# The package name.
PACKAGE=plt

# Path to temporary storage.
if [ "$TMP" = "" ]
then
    TMP=/tmp
fi

# PLT src directory.
PLT=$PWD

# Create a temporary local RPM build environment.
mkdir -p --verbose $TMP/$PACKAGE-rpms/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
mkdir -p --verbose $TMP/$PACKAGE-rpms/RPMS/{athlon,i386,i486,i586,i686,noarch}
# Preserve the user's .rpmmacros file.
mv -f ~/.rpmmacros ~/.rpmmacros.save && /dev/null
echo "%_topdir $TMP/$PACKAGE-rpms" > ~/.rpmmacros

# Tar up the src files.
echo "Creating a tarball of plt."
cd ../../
tar -czf $TMP/$PACKAGE-rpms/SOURCES/plt.src.x.tar.gz plt

# Build the RPMs.
cd $PLT
rpmbuild -ba --clean plt.spec

# Put the RPMs in this directory.
mv -f $TMP/$PACKAGE-rpms/RPMS/athlon/* .
mv -f $TMP/$PACKAGE-rpms/RPMS/i386/* .
mv -f $TMP/$PACKAGE-rpms/RPMS/i486/* .

```

```
mv -f $TMP/$PACKAGE-rpms/RPMS/i586/* .
mv -f $TMP/$PACKAGE-rpms/RPMS/i686/* .
mv -f $TMP/$PACKAGE-rpms/RPMS/noarch/* .
mv -f $TMP/$PACKAGE-rpms/SRPMS/* .

# Remove the RPM build environment.
rm -fr $TMP/$PACKAGE-rpms

# Restore the user's .rpmmacros file if they had one.
rm -f ~/.rpmmacros
mv -f ~/.rpmmacros.save ~/.rpmmacros &> /dev/null
```

### **Lessons Learned/Problems**

I learned that there are more than a dozen ways to create an RPM. I realized that it is best to make the spec file as simple as possible so that users other than yourself can maintain it for future releases.

The biggest problem I had resulted from the PLT project's lack of regard for standards. RPM is designed to make the most of the GNU standard of compiling software. PLT did not strictly follow that standard, and as such, I had to be creative and think of a work around.

### **Final Results/Recommendations**

I did manage to create an RPM. I also created the necessary files to enable others to create a PLT RPM for their system with just one command. I would recommend that someone modify the PLT code so that it follows the GNU standard, provided that the developers would allow it.

### **References**

DrScheme. 30 April 2003 <<http://www.drscheme.org/>>

**Project:** net use / cusrmgr Command Builder

**Author:** Trey Buck

### Problem Background

According to ETSU's Office of Information Technology, the *net use* and *cusrmgr* commands are of great use when configuring machines remotely. However, some of the analysts do not take full advantage of these commands as they can be somewhat complex. Since *net use* and *cusrmgr* are often used concurrently, it made sense to create a GUI applet that implements these commands.

### Project Goals

The applet is designed to meet two goals. The first is providing a graphical user interface for running the two commands. This reduces the burden of having to remember what switches to use and checking the syntax when they are forgotten. Second, it serves to bring both commands together in one interface, providing improved ease of use and functionality.

### Source Code and Documentation

The user enters the relevant information in the appropriate fields, clicks the Execute button, and the applet runs the command. The applet does not make system calls to achieve this. It builds a text string containing the appropriate command and switches, invokes a shell, and executes the command on behalf of the user.

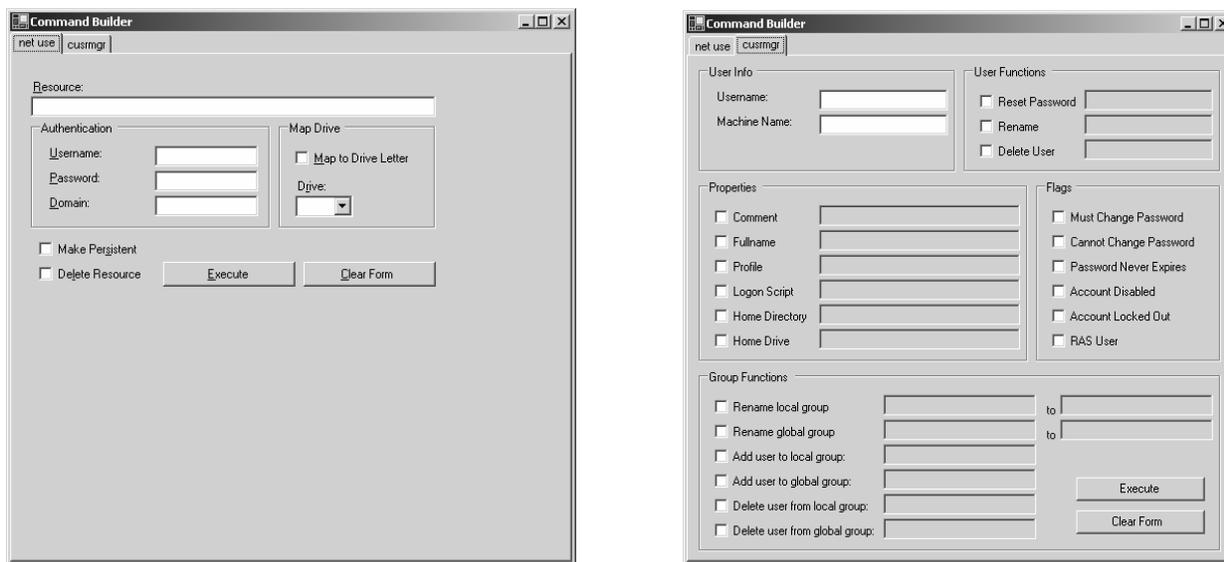


Figure 4.1: Screen shots of the application

The application was written in Microsoft Visual Basic .NET. The Microsoft .NET Framework must be installed on the machine to use the software. The source code is included below:

```
Option Explicit On

Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents TabControl1 As System.Windows.Forms.TabControl
    Friend WithEvents NetPage As System.Windows.Forms.TabPage
    Friend WithEvents UsrPage As System.Windows.Forms.TabPage
    Friend WithEvents lblResource As System.Windows.Forms.Label
    Friend WithEvents txtResource As System.Windows.Forms.TextBox
    Friend WithEvents grpAuthentication As System.Windows.Forms.GroupBox
    Friend WithEvents lblDomain As System.Windows.Forms.Label
    Friend WithEvents lblPassword As System.Windows.Forms.Label
    Friend WithEvents lblUsername As System.Windows.Forms.Label
    Friend WithEvents txtDomain As System.Windows.Forms.TextBox
    Friend WithEvents txtPassword As System.Windows.Forms.TextBox
    Friend WithEvents grpMap As System.Windows.Forms.GroupBox
    Friend WithEvents lblDrive As System.Windows.Forms.Label
    Friend WithEvents chkMap As System.Windows.Forms.CheckBox
    Friend WithEvents chkPersistent As System.Windows.Forms.CheckBox
    Friend WithEvents chkDelete As System.Windows.Forms.CheckBox
    Friend WithEvents cmbDrive As System.Windows.Forms.ComboBox
    Friend WithEvents grpUserInfo As System.Windows.Forms.GroupBox
    Friend WithEvents lblUsername2 As System.Windows.Forms.Label
    Friend WithEvents lblMachineName As System.Windows.Forms.Label
```

```

Friend WithEvents txtMachineName As System.Windows.Forms.TextBox
Friend WithEvents grpUserFunctions As System.Windows.Forms.GroupBox
Friend WithEvents grpGroupFunctions As System.Windows.Forms.GroupBox
Friend WithEvents chkResetPass As System.Windows.Forms.CheckBox
Friend WithEvents txtResetPass As System.Windows.Forms.TextBox
Friend WithEvents chkDeleteUser As System.Windows.Forms.CheckBox
Friend WithEvents chkRenameLocal As System.Windows.Forms.CheckBox
Friend WithEvents chkRenameGlobal As System.Windows.Forms.CheckBox
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents Label2 As System.Windows.Forms.Label
Friend WithEvents grpProperties As System.Windows.Forms.GroupBox
Friend WithEvents txtDeleteUser As System.Windows.Forms.TextBox
Friend WithEvents chkComment As System.Windows.Forms.CheckBox
Friend WithEvents chkFullname As System.Windows.Forms.CheckBox
Friend WithEvents chkProfile As System.Windows.Forms.CheckBox
Friend WithEvents chkLogonScript As System.Windows.Forms.CheckBox
Friend WithEvents chkHomeDirectory As System.Windows.Forms.CheckBox
Friend WithEvents chkHomeDirDrive As System.Windows.Forms.CheckBox
Friend WithEvents txtComment As System.Windows.Forms.TextBox
Friend WithEvents txtFullname As System.Windows.Forms.TextBox
Friend WithEvents txtProfile As System.Windows.Forms.TextBox
Friend WithEvents txtLogonScript As System.Windows.Forms.TextBox
Friend WithEvents txtHomeDirectory As System.Windows.Forms.TextBox
Friend WithEvents txtHomeDirDrive As System.Windows.Forms.TextBox
Friend WithEvents grpFlags As System.Windows.Forms.GroupBox
Friend WithEvents chkMustChangePass As System.Windows.Forms.CheckBox
Friend WithEvents chkCannotChangePass As System.Windows.Forms.CheckBox
Friend WithEvents chkPassNeverExpires As System.Windows.Forms.CheckBox
Friend WithEvents chkDisabled As System.Windows.Forms.CheckBox
Friend WithEvents chkLockedOut As System.Windows.Forms.CheckBox
Friend WithEvents chkRASUser As System.Windows.Forms.CheckBox
Friend WithEvents chkAddGlobal As System.Windows.Forms.CheckBox
Friend WithEvents txtDelGlobal As System.Windows.Forms.TextBox
Friend WithEvents txtDelLocal As System.Windows.Forms.TextBox
Friend WithEvents txtAddGlobal As System.Windows.Forms.TextBox
Friend WithEvents txtAddLocal As System.Windows.Forms.TextBox
Friend WithEvents chkAddLocal As System.Windows.Forms.CheckBox
Friend WithEvents chkDelLocal As System.Windows.Forms.CheckBox
Friend WithEvents btnExecuteNet As System.Windows.Forms.Button
Friend WithEvents btnClearNet As System.Windows.Forms.Button
Friend WithEvents txtUsernameNet As System.Windows.Forms.TextBox
Friend WithEvents txtUsernameMgr As System.Windows.Forms.TextBox
Friend WithEvents btnExecuteMgr As System.Windows.Forms.Button
Friend WithEvents btnClearMgr As System.Windows.Forms.Button
Friend WithEvents txtOldGlobal As System.Windows.Forms.TextBox
Friend WithEvents txtNewLocal As System.Windows.Forms.TextBox
Friend WithEvents txtOldLocal As System.Windows.Forms.TextBox
Friend WithEvents txtNewGlobal As System.Windows.Forms.TextBox
Friend WithEvents chkDelGlobal As System.Windows.Forms.CheckBox
Friend WithEvents txtRename As System.Windows.Forms.TextBox
Friend WithEvents chkRename As System.Windows.Forms.CheckBox
<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
    Me.TabControl1 = New System.Windows.Forms.TabControl()
    Me.NetPage = New System.Windows.Forms.TabPage()
    Me.btnExecuteNet = New System.Windows.Forms.Button()
    Me.btnClearNet = New System.Windows.Forms.Button()

```

```

Me.chkPersistent = New System.Windows.Forms.CheckBox()
Me.grpAuthentication = New System.Windows.Forms.GroupBox()
Me.lblDomain = New System.Windows.Forms.Label()
Me.lblPassword = New System.Windows.Forms.Label()
Me.lblUsername = New System.Windows.Forms.Label()
Me.txtDomain = New System.Windows.Forms.TextBox()
Me.txtPassword = New System.Windows.Forms.TextBox()
Me.txtUsernameNet = New System.Windows.Forms.TextBox()
Me.txtResource = New System.Windows.Forms.TextBox()
Me.lblResource = New System.Windows.Forms.Label()
Me.grpMap = New System.Windows.Forms.GroupBox()
Me.lblDrive = New System.Windows.Forms.Label()
Me.chkMap = New System.Windows.Forms.CheckBox()
Me.cmbDrive = New System.Windows.Forms.ComboBox()
Me.chkDelete = New System.Windows.Forms.CheckBox()
Me.usrPage = New System.Windows.Forms.TabPage()
Me.grpFlags = New System.Windows.Forms.GroupBox()
Me.chkRASUser = New System.Windows.Forms.CheckBox()
Me.chkLockedOut = New System.Windows.Forms.CheckBox()
Me.chkDisabled = New System.Windows.Forms.CheckBox()
Me.chkPassNeverExpires = New System.Windows.Forms.CheckBox()
Me.chkCannotChangePass = New System.Windows.Forms.CheckBox()
Me.chkMustChangePass = New System.Windows.Forms.CheckBox()
Me.grpGroupFunctions = New System.Windows.Forms.GroupBox()
Me.btnClearMgr = New System.Windows.Forms.Button()
Me.btnExecuteMgr = New System.Windows.Forms.Button()
Me.chkAddGlobal = New System.Windows.Forms.CheckBox()
Me.txtDelGlobal = New System.Windows.Forms.TextBox()
Me.txtDelLocal = New System.Windows.Forms.TextBox()
Me.txtAddGlobal = New System.Windows.Forms.TextBox()
Me.txtAddLocal = New System.Windows.Forms.TextBox()
Me.chkAddLocal = New System.Windows.Forms.CheckBox()
Me.chkDelGlobal = New System.Windows.Forms.CheckBox()
Me.chkDelLocal = New System.Windows.Forms.CheckBox()
Me.Label2 = New System.Windows.Forms.Label()
Me.Label1 = New System.Windows.Forms.Label()
Me.txtNewGlobal = New System.Windows.Forms.TextBox()
Me.txtOldGlobal = New System.Windows.Forms.TextBox()
Me.txtNewLocal = New System.Windows.Forms.TextBox()
Me.txtOldLocal = New System.Windows.Forms.TextBox()
Me.chkRenameGlobal = New System.Windows.Forms.CheckBox()
Me.chkRenameLocal = New System.Windows.Forms.CheckBox()
Me.grpUserFunctions = New System.Windows.Forms.GroupBox()
Me.txtRename = New System.Windows.Forms.TextBox()
Me.chkRename = New System.Windows.Forms.CheckBox()
Me.txtDeleteUser = New System.Windows.Forms.TextBox()
Me.chkDeleteUser = New System.Windows.Forms.CheckBox()
Me.txtResetPass = New System.Windows.Forms.TextBox()
Me.chkResetPass = New System.Windows.Forms.CheckBox()
Me.grpUserInfo = New System.Windows.Forms.GroupBox()
Me.txtMachineName = New System.Windows.Forms.TextBox()
Me.txtUsernameMgr = New System.Windows.Forms.TextBox()
Me.lblMachineName = New System.Windows.Forms.Label()
Me.lblUsername2 = New System.Windows.Forms.Label()
Me.grpProperties = New System.Windows.Forms.GroupBox()
Me.txtHomeDirDrive = New System.Windows.Forms.TextBox()
Me.txtHomeDirectory = New System.Windows.Forms.TextBox()

```

```

Me.txtLogonScript = New System.Windows.Forms.TextBox()
Me.txtProfile = New System.Windows.Forms.TextBox()
Me.txtFullname = New System.Windows.Forms.TextBox()
Me.txtComment = New System.Windows.Forms.TextBox()
Me.chkHomeDirDrive = New System.Windows.Forms.CheckBox()
Me.chkHomeDirectory = New System.Windows.Forms.CheckBox()
Me.chkLogonScript = New System.Windows.Forms.CheckBox()
Me.chkProfile = New System.Windows.Forms.CheckBox()
Me.chkFullname = New System.Windows.Forms.CheckBox()
Me.chkComment = New System.Windows.Forms.CheckBox()
Me.TabControl1.SuspendLayout()
Me.NetPage.SuspendLayout()
Me.grpAuthentication.SuspendLayout()
Me.grpMap.SuspendLayout()
Me.UsrPage.SuspendLayout()
Me.grpFlags.SuspendLayout()
Me.grpGroupFunctions.SuspendLayout()
Me.grpUserFunctions.SuspendLayout()
Me.grpUserInfo.SuspendLayout()
Me.grpProperties.SuspendLayout()
Me.SuspendLayout()
'
'TabControl1
'
Me.TabControl1.Controls.AddRange(New System.Windows.Forms.Control()
{Me.NetPage, Me.UsrPage})
Me.TabControl1.Name = "TabControl1"
Me.TabControl1.SelectedIndex = 0
Me.TabControl1.Size = New System.Drawing.Size(544, 512)
Me.TabControl1.TabIndex = 0
'
'NetPage
'
Me.NetPage.Controls.AddRange(New System.Windows.Forms.Control()
{Me.btnExecuteNet, Me.btnCancelNet, Me.chkPersistent, Me.grpAuthentication,
Me.txtResource, Me.lblResource, Me.grpMap, Me.chkDelete})
Me.NetPage.Location = New System.Drawing.Point(4, 22)
Me.NetPage.Name = "NetPage"
Me.NetPage.Size = New System.Drawing.Size(536, 486)
Me.NetPage.TabIndex = 0
Me.NetPage.Text = "net use"
'
'btnExecuteNet
'
Me.btnExecuteNet.Location = New System.Drawing.Point(144, 200)
Me.btnExecuteNet.Name = "btnExecuteNet"
Me.btnExecuteNet.Size = New System.Drawing.Size(128, 24)
Me.btnExecuteNet.TabIndex = 6
Me.btnExecuteNet.Text = "&Execute"
'
'btnClearNet
'
Me.btnCancelNet.Location = New System.Drawing.Point(280, 200)
Me.btnCancelNet.Name = "btnClearNet"
Me.btnCancelNet.Size = New System.Drawing.Size(128, 24)
Me.btnCancelNet.TabIndex = 7
Me.btnCancelNet.Text = "&Clear Form"

```

```

'
'chkPersistent
'
Me.chkPersistent.Location = New System.Drawing.Point(24, 176)
Me.chkPersistent.Name = "chkPersistent"
Me.chkPersistent.Size = New System.Drawing.Size(112, 24)
Me.chkPersistent.TabIndex = 4
Me.chkPersistent.Text = "Make Per&sistent"
'
'grpAuthentication
'
Me.grpAuthentication.Controls.AddRange(New
System.Windows.Forms.Control() {Me.lblDomain, Me.lblPassword, Me.lblUsername,
Me.txtDomain, Me.txtPassword, Me.txtUsernameNet})
Me.grpAuthentication.Location = New System.Drawing.Point(16, 64)
Me.grpAuthentication.Name = "grpAuthentication"
Me.grpAuthentication.Size = New System.Drawing.Size(232, 104)
Me.grpAuthentication.TabIndex = 2
Me.grpAuthentication.TabStop = False
Me.grpAuthentication.Text = "Authentication"
'
'lblDomain
'
Me.lblDomain.Location = New System.Drawing.Point(16, 72)
Me.lblDomain.Name = "lblDomain"
Me.lblDomain.Size = New System.Drawing.Size(100, 16)
Me.lblDomain.TabIndex = 4
Me.lblDomain.Text = "&Domain:"
'
'lblPassword
'
Me.lblPassword.Location = New System.Drawing.Point(16, 48)
Me.lblPassword.Name = "lblPassword"
Me.lblPassword.Size = New System.Drawing.Size(100, 16)
Me.lblPassword.TabIndex = 2
Me.lblPassword.Text = "&Password:"
'
'lblUsername
'
Me.lblUsername.Location = New System.Drawing.Point(16, 24)
Me.lblUsername.Name = "lblUsername"
Me.lblUsername.Size = New System.Drawing.Size(100, 16)
Me.lblUsername.TabIndex = 0
Me.lblUsername.Text = "&Username:"
'
'txtDomain
'
Me.txtDomain.Location = New System.Drawing.Point(120, 72)
Me.txtDomain.Name = "txtDomain"
Me.txtDomain.TabIndex = 5
Me.txtDomain.Text = ""
'
'txtPassword
'
Me.txtPassword.Location = New System.Drawing.Point(120, 48)
Me.txtPassword.Name = "txtPassword"
Me.txtPassword.PasswordChar = Microsoft.VisualBasic.ChrW(42)

```

```

Me.txtPassword.TabIndex = 3
Me.txtPassword.Text = ""
'
'txtUsernameNet
'
Me.txtUsernameNet.Location = New System.Drawing.Point(120, 24)
Me.txtUsernameNet.Name = "txtUsernameNet"
Me.txtUsernameNet.TabIndex = 1
Me.txtUsernameNet.Text = ""
'
'txtResource
'
Me.txtResource.Location = New System.Drawing.Point(16, 40)
Me.txtResource.Name = "txtResource"
Me.txtResource.Size = New System.Drawing.Size(392, 20)
Me.txtResource.TabIndex = 1
Me.txtResource.Text = ""
'
'lblResource
'
Me.lblResource.Location = New System.Drawing.Point(16, 24)
Me.lblResource.Name = "lblResource"
Me.lblResource.Size = New System.Drawing.Size(104, 16)
Me.lblResource.TabIndex = 0
Me.lblResource.Text = "&Resource:"
'
'grpMap
'
Me.grpMap.Controls.AddRange(New System.Windows.Forms.Control()
{Me.lblDrive, Me.chkMap, Me.cmbDrive})
Me.grpMap.Location = New System.Drawing.Point(256, 64)
Me.grpMap.Name = "grpMap"
Me.grpMap.Size = New System.Drawing.Size(152, 104)
Me.grpMap.TabIndex = 3
Me.grpMap.TabStop = False
Me.grpMap.Text = "Map Drive"
'
'lblDrive
'
Me.lblDrive.Location = New System.Drawing.Point(16, 56)
Me.lblDrive.Name = "lblDrive"
Me.lblDrive.Size = New System.Drawing.Size(100, 16)
Me.lblDrive.TabIndex = 1
Me.lblDrive.Text = "D&rive:"
'
'chkMap
'
Me.chkMap.Location = New System.Drawing.Point(16, 24)
Me.chkMap.Name = "chkMap"
Me.chkMap.Size = New System.Drawing.Size(128, 24)
Me.chkMap.TabIndex = 0
Me.chkMap.Text = "&Map to Drive Letter"
'
'cmbDrive
'

```

```

        Me.cmbDrive.Items.AddRange(New Object() {"D:", "E:", "F:", "G:", "H:",
"I:", "J:", "K:", "L:", "M:", "N:", "O:", "P:", "Q:", "R:", "S:", "T:", "U:",
"V:", "W:", "X:", "Y:", "Z:"})
        Me.cmbDrive.Location = New System.Drawing.Point(16, 72)
        Me.cmbDrive.Name = "cmbDrive"
        Me.cmbDrive.Size = New System.Drawing.Size(56, 21)
        Me.cmbDrive.TabIndex = 2
        '
        'chkDelete
        '
        Me.chkDelete.Location = New System.Drawing.Point(24, 200)
        Me.chkDelete.Name = "chkDelete"
        Me.chkDelete.Size = New System.Drawing.Size(120, 24)
        Me.chkDelete.TabIndex = 5
        Me.chkDelete.Text = "De&lete Resource"
        '
        'UsrPage
        '
        Me.UsrPage.Controls.AddRange(New System.Windows.Forms.Control()
{Me.grpFlags, Me.grpGroupFunctions, Me.grpUserFunctions, Me.grpUserInfo,
Me.grpProperties})
        Me.UsrPage.Location = New System.Drawing.Point(4, 22)
        Me.UsrPage.Name = "UsrPage"
        Me.UsrPage.Size = New System.Drawing.Size(536, 486)
        Me.UsrPage.TabIndex = 1
        Me.UsrPage.Text = "cusrmgr"
        '
        'grpFlags
        '
        Me.grpFlags.Controls.AddRange(New System.Windows.Forms.Control()
{Me.chkRASUser, Me.chkLockedOut, Me.chkDisabled, Me.chkPassNeverExpires,
Me.chkCannotChangePass, Me.chkMustChangePass})
        Me.grpFlags.Location = New System.Drawing.Point(344, 120)
        Me.grpFlags.Name = "grpFlags"
        Me.grpFlags.Size = New System.Drawing.Size(184, 176)
        Me.grpFlags.TabIndex = 3
        Me.grpFlags.TabStop = False
        Me.grpFlags.Text = "Flags"
        '
        'chkRASUser
        '
        Me.chkRASUser.Location = New System.Drawing.Point(16, 144)
        Me.chkRASUser.Name = "chkRASUser"
        Me.chkRASUser.Size = New System.Drawing.Size(160, 24)
        Me.chkRASUser.TabIndex = 5
        Me.chkRASUser.Text = "RAS User"
        '
        'chkLockedOut
        '
        Me.chkLockedOut.Location = New System.Drawing.Point(16, 120)
        Me.chkLockedOut.Name = "chkLockedOut"
        Me.chkLockedOut.Size = New System.Drawing.Size(160, 24)
        Me.chkLockedOut.TabIndex = 4
        Me.chkLockedOut.Text = "Account Locked Out"
        '
        'chkDisabled
        '

```

```

Me.chkDisabled.Location = New System.Drawing.Point(16, 96)
Me.chkDisabled.Name = "chkDisabled"
Me.chkDisabled.Size = New System.Drawing.Size(160, 24)
Me.chkDisabled.TabIndex = 3
Me.chkDisabled.Text = "Account Disabled"
'
'chkPassNeverExpires
'
Me.chkPassNeverExpires.Location = New System.Drawing.Point(16, 72)
Me.chkPassNeverExpires.Name = "chkPassNeverExpires"
Me.chkPassNeverExpires.Size = New System.Drawing.Size(160, 24)
Me.chkPassNeverExpires.TabIndex = 2
Me.chkPassNeverExpires.Text = "Password Never Expires"
'
'chkCannotChangePass
'
Me.chkCannotChangePass.Location = New System.Drawing.Point(16, 48)
Me.chkCannotChangePass.Name = "chkCannotChangePass"
Me.chkCannotChangePass.Size = New System.Drawing.Size(160, 24)
Me.chkCannotChangePass.TabIndex = 1
Me.chkCannotChangePass.Text = "Cannot Change Password"
'
'chkMustChangePass
'
Me.chkMustChangePass.Location = New System.Drawing.Point(16, 24)
Me.chkMustChangePass.Name = "chkMustChangePass"
Me.chkMustChangePass.Size = New System.Drawing.Size(160, 24)
Me.chkMustChangePass.TabIndex = 0
Me.chkMustChangePass.Text = "Must Change Password"
'
'grpGroupFunctions
'
Me.grpGroupFunctions.Controls.AddRange(New
System.Windows.Forms.Control() {Me.btnClearMgr, Me.btnExecuteMgr,
Me.chkAddGlobal, Me.txtDelGlobal, Me.txtDelLocal, Me.txtAddGlobal,
Me.txtAddLocal, Me.chkAddLocal, Me.chkDelGlobal, Me.chkDelLocal, Me.Label2,
Me.Label1, Me.txtNewGlobal, Me.txtOldGlobal, Me.txtNewLocal, Me.txtOldLocal,
Me.chkRenameGlobal, Me.chkRenameLocal})
Me.grpGroupFunctions.Location = New System.Drawing.Point(8, 304)
Me.grpGroupFunctions.Name = "grpGroupFunctions"
Me.grpGroupFunctions.Size = New System.Drawing.Size(520, 176)
Me.grpGroupFunctions.TabIndex = 4
Me.grpGroupFunctions.TabStop = False
Me.grpGroupFunctions.Text = "Group Functions"
'
'btnClearMgr
'
Me.btnClearMgr.Location = New System.Drawing.Point(376, 136)
Me.btnClearMgr.Name = "btnClearMgr"
Me.btnClearMgr.Size = New System.Drawing.Size(128, 23)
Me.btnClearMgr.TabIndex = 17
Me.btnClearMgr.Text = "Clear Form"
'
'btnExecuteMgr
'
Me.btnExecuteMgr.Location = New System.Drawing.Point(376, 104)
Me.btnExecuteMgr.Name = "btnExecuteMgr"

```

```

Me.btnExecuteMgr.Size = New System.Drawing.Size(128, 23)
Me.btnExecuteMgr.TabIndex = 16
Me.btnExecuteMgr.Text = "Execute"
'
'chkAddGlobal
'
Me.chkAddGlobal.Location = New System.Drawing.Point(16, 96)
Me.chkAddGlobal.Name = "chkAddGlobal"
Me.chkAddGlobal.Size = New System.Drawing.Size(152, 24)
Me.chkAddGlobal.TabIndex = 10
Me.chkAddGlobal.Text = "Add user to global group:"
'
'txtDelGlobal
'
Me.txtDelGlobal.Enabled = False
Me.txtDelGlobal.Location = New System.Drawing.Point(184, 144)
Me.txtDelGlobal.Name = "txtDelGlobal"
Me.txtDelGlobal.Size = New System.Drawing.Size(152, 20)
Me.txtDelGlobal.TabIndex = 15
Me.txtDelGlobal.Text = ""
'
'txtDelLocal
'
Me.txtDelLocal.Enabled = False
Me.txtDelLocal.Location = New System.Drawing.Point(184, 120)
Me.txtDelLocal.Name = "txtDelLocal"
Me.txtDelLocal.Size = New System.Drawing.Size(152, 20)
Me.txtDelLocal.TabIndex = 13
Me.txtDelLocal.Text = ""
'
'txtAddGlobal
'
Me.txtAddGlobal.Enabled = False
Me.txtAddGlobal.Location = New System.Drawing.Point(184, 96)
Me.txtAddGlobal.Name = "txtAddGlobal"
Me.txtAddGlobal.Size = New System.Drawing.Size(152, 20)
Me.txtAddGlobal.TabIndex = 11
Me.txtAddGlobal.Text = ""
'
'txtAddLocal
'
Me.txtAddLocal.Enabled = False
Me.txtAddLocal.Location = New System.Drawing.Point(184, 72)
Me.txtAddLocal.Name = "txtAddLocal"
Me.txtAddLocal.Size = New System.Drawing.Size(152, 20)
Me.txtAddLocal.TabIndex = 9
Me.txtAddLocal.Text = ""
'
'chkAddLocal
'
Me.chkAddLocal.Location = New System.Drawing.Point(16, 72)
Me.chkAddLocal.Name = "chkAddLocal"
Me.chkAddLocal.Size = New System.Drawing.Size(152, 24)
Me.chkAddLocal.TabIndex = 8
Me.chkAddLocal.Text = "Add user to local group:"
'
'chkDelGlobal

```

```

'
Me.chkDelGlobal.Location = New System.Drawing.Point(16, 144)
Me.chkDelGlobal.Name = "chkDelGlobal"
Me.chkDelGlobal.Size = New System.Drawing.Size(176, 24)
Me.chkDelGlobal.TabIndex = 14
Me.chkDelGlobal.Text = "Delete user from global group:"
'
'chkDelLocal
'
Me.chkDelLocal.Location = New System.Drawing.Point(16, 120)
Me.chkDelLocal.Name = "chkDelLocal"
Me.chkDelLocal.Size = New System.Drawing.Size(168, 24)
Me.chkDelLocal.TabIndex = 12
Me.chkDelLocal.Text = "Delete user from local group:"
'
'Label2
'
Me.Label2.Location = New System.Drawing.Point(344, 32)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(16, 16)
Me.Label2.TabIndex = 2
Me.Label2.Text = "to:"
'
'Label1
'
Me.Label1.Location = New System.Drawing.Point(344, 56)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(16, 16)
Me.Label1.TabIndex = 6
Me.Label1.Text = "to:"
'
'txtNewGlobal
'
Me.txtNewGlobal.Enabled = False
Me.txtNewGlobal.Location = New System.Drawing.Point(360, 48)
Me.txtNewGlobal.Name = "txtNewGlobal"
Me.txtNewGlobal.Size = New System.Drawing.Size(152, 20)
Me.txtNewGlobal.TabIndex = 7
Me.txtNewGlobal.Text = ""
'
'txtOldGlobal
'
Me.txtOldGlobal.Enabled = False
Me.txtOldGlobal.Location = New System.Drawing.Point(184, 48)
Me.txtOldGlobal.Name = "txtOldGlobal"
Me.txtOldGlobal.Size = New System.Drawing.Size(152, 20)
Me.txtOldGlobal.TabIndex = 5
Me.txtOldGlobal.Text = ""
'
'txtNewLocal
'
Me.txtNewLocal.Enabled = False
Me.txtNewLocal.Location = New System.Drawing.Point(360, 24)
Me.txtNewLocal.Name = "txtNewLocal"
Me.txtNewLocal.Size = New System.Drawing.Size(152, 20)
Me.txtNewLocal.TabIndex = 3
Me.txtNewLocal.Text = ""

```

```

'
'txtOldLocal
'
Me.txtOldLocal.Enabled = False
Me.txtOldLocal.Location = New System.Drawing.Point(184, 24)
Me.txtOldLocal.Name = "txtOldLocal"
Me.txtOldLocal.Size = New System.Drawing.Size(152, 20)
Me.txtOldLocal.TabIndex = 1
Me.txtOldLocal.Text = ""
'
'chkRenameGlobal
'
Me.chkRenameGlobal.Location = New System.Drawing.Point(16, 48)
Me.chkRenameGlobal.Name = "chkRenameGlobal"
Me.chkRenameGlobal.Size = New System.Drawing.Size(136, 24)
Me.chkRenameGlobal.TabIndex = 4
Me.chkRenameGlobal.Text = "Rename global group"
'
'chkRenameLocal
'
Me.chkRenameLocal.Location = New System.Drawing.Point(16, 24)
Me.chkRenameLocal.Name = "chkRenameLocal"
Me.chkRenameLocal.Size = New System.Drawing.Size(136, 24)
Me.chkRenameLocal.TabIndex = 0
Me.chkRenameLocal.Text = "Rename local group"
'
'grpUserFunctions
'
Me.grpUserFunctions.Controls.AddRange(New
System.Windows.Forms.Control() {Me.txtRename, Me.chkRename, Me.txtDeleteUser,
Me.chkDeleteUser, Me.txtResetPass, Me.chkResetPass})
Me.grpUserFunctions.Location = New System.Drawing.Point(272, 8)
Me.grpUserFunctions.Name = "grpUserFunctions"
Me.grpUserFunctions.Size = New System.Drawing.Size(256, 104)
Me.grpUserFunctions.TabIndex = 1
Me.grpUserFunctions.TabStop = False
Me.grpUserFunctions.Text = "User Functions"
'
'txtRename
'
Me.txtRename.Enabled = False
Me.txtRename.Location = New System.Drawing.Point(120, 48)
Me.txtRename.Name = "txtRename"
Me.txtRename.Size = New System.Drawing.Size(128, 20)
Me.txtRename.TabIndex = 7
Me.txtRename.Text = ""
'
'chkRename
'
Me.chkRename.Location = New System.Drawing.Point(16, 48)
Me.chkRename.Name = "chkRename"
Me.chkRename.Size = New System.Drawing.Size(80, 24)
Me.chkRename.TabIndex = 6
Me.chkRename.Text = "Rename"
'
'txtDeleteUser
'

```

```

Me.txtDeleteUser.Enabled = False
Me.txtDeleteUser.Location = New System.Drawing.Point(120, 72)
Me.txtDeleteUser.Name = "txtDeleteUser"
Me.txtDeleteUser.Size = New System.Drawing.Size(128, 20)
Me.txtDeleteUser.TabIndex = 5
Me.txtDeleteUser.Text = ""
'
'chkDeleteUser
'
Me.chkDeleteUser.Location = New System.Drawing.Point(16, 72)
Me.chkDeleteUser.Name = "chkDeleteUser"
Me.chkDeleteUser.TabIndex = 4
Me.chkDeleteUser.Text = "Delete User"
'
'txtResetPass
'
Me.txtResetPass.Enabled = False
Me.txtResetPass.Location = New System.Drawing.Point(120, 24)
Me.txtResetPass.Name = "txtResetPass"
Me.txtResetPass.Size = New System.Drawing.Size(128, 20)
Me.txtResetPass.TabIndex = 3
Me.txtResetPass.Text = ""
'
'chkResetPass
'
Me.chkResetPass.Location = New System.Drawing.Point(16, 24)
Me.chkResetPass.Name = "chkResetPass"
Me.chkResetPass.Size = New System.Drawing.Size(112, 24)
Me.chkResetPass.TabIndex = 2
Me.chkResetPass.Text = "Reset Password"
'
'grpUserInfo
'
Me.grpUserInfo.Controls.AddRange(New System.Windows.Forms.Control()
{Me.txtMachineName, Me.txtUsernameMgr, Me.lblMachineName, Me.lblUsername2})
Me.grpUserInfo.Location = New System.Drawing.Point(8, 8)
Me.grpUserInfo.Name = "grpUserInfo"
Me.grpUserInfo.Size = New System.Drawing.Size(256, 104)
Me.grpUserInfo.TabIndex = 0
Me.grpUserInfo.TabStop = False
Me.grpUserInfo.Text = "User Info"
'
'txtMachineName
'
Me.txtMachineName.Location = New System.Drawing.Point(120, 48)
Me.txtMachineName.Name = "txtMachineName"
Me.txtMachineName.Size = New System.Drawing.Size(128, 20)
Me.txtMachineName.TabIndex = 3
Me.txtMachineName.Text = ""
'
'txtUsernameMgr
'
Me.txtUsernameMgr.Location = New System.Drawing.Point(120, 24)
Me.txtUsernameMgr.Name = "txtUsernameMgr"
Me.txtUsernameMgr.Size = New System.Drawing.Size(128, 20)
Me.txtUsernameMgr.TabIndex = 1
Me.txtUsernameMgr.Text = ""

```

```

'
'lblMachineName
'
Me.lblMachineName.Location = New System.Drawing.Point(16, 48)
Me.lblMachineName.Name = "lblMachineName"
Me.lblMachineName.Size = New System.Drawing.Size(104, 16)
Me.lblMachineName.TabIndex = 2
Me.lblMachineName.Text = "Machine Name:"
'
'lblUsername2
'
Me.lblUsername2.Location = New System.Drawing.Point(16, 24)
Me.lblUsername2.Name = "lblUsername2"
Me.lblUsername2.Size = New System.Drawing.Size(100, 16)
Me.lblUsername2.TabIndex = 0
Me.lblUsername2.Text = "Username:"
'
'grpProperties
'
Me.grpProperties.Controls.AddRange(New System.Windows.Forms.Control()
{Me.txtHomeDirDrive, Me.txtHomeDirectory, Me.txtLogonScript, Me.txtProfile,
Me.txtFullname, Me.txtComment, Me.chkHomeDirDrive, Me.chkHomeDirectory,
Me.chkLogonScript, Me.chkProfile, Me.chkFullname, Me.chkComment})
Me.grpProperties.Location = New System.Drawing.Point(8, 120)
Me.grpProperties.Name = "grpProperties"
Me.grpProperties.Size = New System.Drawing.Size(328, 176)
Me.grpProperties.TabIndex = 2
Me.grpProperties.TabStop = False
Me.grpProperties.Text = "Properties"
'
'txtHomeDirDrive
'
Me.txtHomeDirDrive.Enabled = False
Me.txtHomeDirDrive.Location = New System.Drawing.Point(120, 144)
Me.txtHomeDirDrive.Name = "txtHomeDirDrive"
Me.txtHomeDirDrive.Size = New System.Drawing.Size(200, 20)
Me.txtHomeDirDrive.TabIndex = 11
Me.txtHomeDirDrive.Text = ""
'
'txtHomeDirectory
'
Me.txtHomeDirectory.Enabled = False
Me.txtHomeDirectory.Location = New System.Drawing.Point(120, 120)
Me.txtHomeDirectory.Name = "txtHomeDirectory"
Me.txtHomeDirectory.Size = New System.Drawing.Size(200, 20)
Me.txtHomeDirectory.TabIndex = 9
Me.txtHomeDirectory.Text = ""
'
'txtLogonScript
'
Me.txtLogonScript.Enabled = False
Me.txtLogonScript.Location = New System.Drawing.Point(120, 96)
Me.txtLogonScript.Name = "txtLogonScript"
Me.txtLogonScript.Size = New System.Drawing.Size(200, 20)
Me.txtLogonScript.TabIndex = 7
Me.txtLogonScript.Text = ""
'

```

```

'txtProfile
,
Me.txtProfile.Enabled = False
Me.txtProfile.Location = New System.Drawing.Point(120, 72)
Me.txtProfile.Name = "txtProfile"
Me.txtProfile.Size = New System.Drawing.Size(200, 20)
Me.txtProfile.TabIndex = 5
Me.txtProfile.Text = ""
,
'txtFullname
,
Me.txtFullname.Enabled = False
Me.txtFullname.Location = New System.Drawing.Point(120, 48)
Me.txtFullname.Name = "txtFullname"
Me.txtFullname.Size = New System.Drawing.Size(200, 20)
Me.txtFullname.TabIndex = 3
Me.txtFullname.Text = ""
,
'txtComment
,
Me.txtComment.Enabled = False
Me.txtComment.Location = New System.Drawing.Point(120, 24)
Me.txtComment.Name = "txtComment"
Me.txtComment.Size = New System.Drawing.Size(200, 20)
Me.txtComment.TabIndex = 1
Me.txtComment.Text = ""
,
'chkHomeDirDrive
,
Me.chkHomeDirDrive.Location = New System.Drawing.Point(16, 144)
Me.chkHomeDirDrive.Name = "chkHomeDirDrive"
Me.chkHomeDirDrive.TabIndex = 10
Me.chkHomeDirDrive.Text = "Home Drive"
,
'chkHomeDirectory
,
Me.chkHomeDirectory.Location = New System.Drawing.Point(16, 120)
Me.chkHomeDirectory.Name = "chkHomeDirectory"
Me.chkHomeDirectory.TabIndex = 8
Me.chkHomeDirectory.Text = "Home Directory"
,
'chkLogonScript
,
Me.chkLogonScript.Location = New System.Drawing.Point(16, 96)
Me.chkLogonScript.Name = "chkLogonScript"
Me.chkLogonScript.TabIndex = 6
Me.chkLogonScript.Text = "Logon Script"
,
'chkProfile
,
Me.chkProfile.Location = New System.Drawing.Point(16, 72)
Me.chkProfile.Name = "chkProfile"
Me.chkProfile.TabIndex = 4
Me.chkProfile.Text = "Profile"
,
'chkFullname
,

```

```

Me.chkFullname.Location = New System.Drawing.Point(16, 48)
Me.chkFullname.Name = "chkFullname"
Me.chkFullname.TabIndex = 2
Me.chkFullname.Text = "Fullname"
'
'chkComment
'
Me.chkComment.Location = New System.Drawing.Point(16, 24)
Me.chkComment.Name = "chkComment"
Me.chkComment.TabIndex = 0
Me.chkComment.Text = "Comment"
'
'Form1
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(544, 517)
Me.Controls.AddRange(New System.Windows.Forms.Control()
{Me.TabControll1})
Me.Name = "Form1"
Me.Text = "Command Builder"
Me.TabControll1.ResumeLayout(False)
Me.NetPage.ResumeLayout(False)
Me.grpAuthentication.ResumeLayout(False)
Me.grpMap.ResumeLayout(False)
Me.usrPage.ResumeLayout(False)
Me.grpFlags.ResumeLayout(False)
Me.grpGroupFunctions.ResumeLayout(False)
Me.grpUserFunctions.ResumeLayout(False)
Me.grpUserInfo.ResumeLayout(False)
Me.grpProperties.ResumeLayout(False)
Me.ResumeLayout(False)

End Sub

#End Region

Private Sub chkDelete_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles chkDelete.CheckedChanged
'
' if deleting a resource, disable other options
'
If chkDelete.Checked = True Then
    grpAuthentication.Enabled = False
    grpMap.Enabled = False
    chkPersistent.Enabled = False
Else
    grpAuthentication.Enabled = True
    grpMap.Enabled = True
    chkPersistent.Enabled = True
End If

End Sub

End Sub

```

```

' clear the net form
'
Private Sub btnClearNet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearNet.Click

    txtResource.Text = ""
    txtUsernameNet.Text = ""
    txtPassword.Text = ""
    txtDomain.Text = ""
    chkMap.Checked = False
    chkPersistent.Checked = False
    chkDelete.Checked = False

End Sub

'
' Build and execute the net command
'
Private Sub btnExecuteNet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExecuteNet.Click

    Dim strCommand As String

    strCommand = "cmd /k net use "

    '
    ' Build the connection string
    '
    If chkDelete.Checked = True Then
        strCommand = strCommand & txtResource.Text & " /DELETE"
    Else

        If chkMap.Checked = True Then
            strCommand = strCommand & cmbDrive.SelectedItem & " "
        End If

        strCommand = strCommand & txtResource.Text & " "

        If txtPassword.Text <> "" Then
            strCommand = strCommand & txtPassword.Text & " "
        End If

        If txtUsernameNet.Text <> "" Then
            strCommand = strCommand & "/USER:"

            If txtDomain.Text <> "" Then
                strCommand = strCommand & txtDomain.Text & "\"
            End If

            strCommand = strCommand & txtUsernameNet.Text & " "
        End If

        If chkPersistent.Checked = True Then
            strCommand = strCommand & "/PERSISTENT:YES"
        Else

```

```

        'strCommand = strCommand & "/PERSISTENT:NO"
    End If

End If

Shell(strCommand, AppWinStyle.NormalFocus, False)

End Sub

'
' Enable/Disable txtboxes based on chkboxes
'
Private Sub chkResetPass_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkResetPass.CheckedChanged
    txtResetPass.Enabled = chkResetPass.Checked
End Sub

Private Sub chkRename_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles chkRename.CheckedChanged
    txtRename.Enabled = chkRename.Checked
    If chkRename.Checked Then
        chkDeleteUser.Checked = False
    End If
End Sub

Private Sub chkDeleteUser_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkDeleteUser.CheckedChanged
    txtDeleteUser.Enabled = chkDeleteUser.Checked
    If chkDeleteUser.Checked Then
        chkRename.Checked = False
    End If
End Sub

Private Sub chkComment_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles chkComment.CheckedChanged
    txtComment.Enabled = chkComment.Checked
End Sub

Private Sub chkFullname_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkFullname.CheckedChanged
    txtFullname.Enabled = chkFullname.Checked
End Sub

Private Sub chkProfile_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles chkProfile.CheckedChanged
    txtProfile.Enabled = chkProfile.Checked
End Sub

Private Sub chkLogonScript_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkLogonScript.CheckedChanged
    txtLogonScript.Enabled = chkLogonScript.Checked
End Sub

Private Sub chkHomeDirectory_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkHomeDirectory.CheckedChanged
    txtHomeDirectory.Enabled = chkHomeDirectory.Checked

```

```

End Sub

Private Sub chkHomeDirDrive_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkHomeDirDrive.CheckedChanged
    txtHomeDirDrive.Enabled = chkHomeDirDrive.Checked
End Sub

Private Sub chkRenameLocal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkRenameLocal.CheckedChanged
    txtOldLocal.Enabled = chkRenameLocal.Checked
    txtNewLocal.Enabled = chkRenameLocal.Checked
End Sub

Private Sub chkRenameGlobal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkRenameGlobal.CheckedChanged
    txtOldGlobal.Enabled = chkRenameGlobal.Checked
    txtNewGlobal.Enabled = chkRenameGlobal.Checked
End Sub

Private Sub chkAddLocal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkAddLocal.CheckedChanged
    txtAddLocal.Enabled = chkAddLocal.Checked
End Sub

Private Sub chkAddGlobal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkAddGlobal.CheckedChanged
    txtAddGlobal.Enabled = chkAddGlobal.Checked
End Sub

Private Sub chkDelLocal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkDelLocal.CheckedChanged
    txtDelLocal.Enabled = chkDelLocal.Checked
End Sub

Private Sub chkDelGlobal_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkDelGlobal.CheckedChanged
    txtDelGlobal.Enabled = chkDelGlobal.Checked
End Sub

'
' Build and execute the mgr command
'
Private Sub btnExecuteMgr_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExecuteMgr.Click

    Dim strCommand As String

    strCommand = "cmd /k cusrmgr "

    If txtUsernameMgr.Text <> "" Then
        strCommand = strCommand & "-u " & txtUsernameMgr.Text & " "
    End If

    If txtMachineName.Text <> "" Then
        strCommand = strCommand & "-m " & txtMachineName.Text & " "
    End If

```

```

If chkRename.Checked Then
    strCommand = strCommand & "-r " & txtRename.Text & " "
End If

If chkDeleteUser.Checked Then
    strCommand = strCommand & "-d " & txtDeleteUser.Text & " "
End If

If chkResetPass.Checked Then
    strCommand = strCommand & "-P " & txtResetPass.Text & " "
End If

If chkRenameLocal.Checked Then
    strCommand = strCommand & "-rlg " & txtOldLocal.Text & " " &
txtNewLocal.Text & " "
End If

If chkRenameGlobal.Checked Then
    strCommand = strCommand & "-rgg " & txtOldGlobal.Text & " " &
txtNewGlobal.Text & " "
End If

If chkAddLocal.Checked Then
    strCommand = strCommand & "-alg " & txtAddLocal.Text & " "
End If

If chkAddGlobal.Checked Then
    strCommand = strCommand & "-agg " & txtAddGlobal.Text & " "
End If

If chkDelLocal.Checked Then
    strCommand = strCommand & "-dlg " & txtDelLocal.Text & " "
End If

If chkDelGlobal.Checked Then
    strCommand = strCommand & "-dgg " & txtDelGlobal.Text & " "
End If

If chkComment.Checked Then
    strCommand = strCommand & "-c " & txtComment.Text & " "
End If

If chkFullname.Checked Then
    strCommand = strCommand & "-f " & txtFullname.Text & " "
End If

If chkProfile.Checked Then
    strCommand = strCommand & "-U " & txtProfile.Text & " "
End If

If chkLogonScript.Checked Then
    strCommand = strCommand & "-n " & txtLogonScript.Text & " "
End If

If chkHomeDirectory.Checked Then
    strCommand = strCommand & "-h " & txtHomeDirectory.Text & " "
End If

```

```

If chkHomeDirDrive.Checked Then
    strCommand = strCommand & "-H " & txtHomeDirDrive.Text & " "
End If

If chkMustChangePass.Checked Then
    strCommand = strCommand & "+s MustChangePassword "
Else
    strCommand = strCommand & "-s MustChangePassword "
End If

If chkCannotChangePass.Checked Then
    strCommand = strCommand & "+s CannotChangePassword "
Else
    strCommand = strCommand & "-s CannotChangePassword "
End If

If chkPassNeverExpires.Checked Then
    strCommand = strCommand & "+s PasswordNeverExpires "
Else
    strCommand = strCommand & "-s PasswordNeverExpires "
End If

If chkDisabled.Checked Then
    strCommand = strCommand & "+s AccountDisabled "
Else
    strCommand = strCommand & "-s AccountDisabled "
End If

If chkLockedOut.Checked Then
    strCommand = strCommand & "+s AccountLockout "
Else
    strCommand = strCommand & "-s AccountLockout "
End If

If chkRASUser.Checked Then
    strCommand = strCommand & "+s RASUser "
Else
    strCommand = strCommand & "-s RASUser "
End If

Shell(strCommand, AppWinStyle.NormalFocus, False)

End Sub

'
' clear the mgr form
'

Private Sub btnClearMgr_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearMgr.Click

    txtUsernameMgr.Text = ""
    txtMachineName.Text = ""
    chkResetPass.Checked = False
    txtResetPass.Text = ""
    chkRename.Checked = False
    txtRename.Text = ""

```

```
chkDeleteUser.Checked = False
txtDeleteUser.Text = ""
chkComment.Checked = False
txtComment.Text = ""
chkFullname.Checked = False
txtFullname.Text = ""
chkProfile.Checked = False
txtProfile.Text = ""
chkLogonScript.Checked = False
txtLogonScript.Text = ""
chkHomeDirectory.Checked = False
txtHomeDirectory.Text = ""
chkHomeDirDrive.Checked = False
txtHomeDirDrive.Text = ""
chkRenameLocal.Checked = False
txtOldLocal.Text = ""
txtNewLocal.Text = ""
chkRenameGlobal.Checked = False
txtOldGlobal.Text = ""
txtNewGlobal.Text = ""
chkAddLocal.Checked = False
txtAddLocal.Text = ""
chkAddGlobal.Checked = False
txtAddGlobal.Text = ""
chkDelLocal.Checked = False
txtDelLocal.Text = ""
chkDelGlobal.Checked = False
txtDelGlobal.Text = ""
chkMustChangePass.Checked = False
chkCannotChangePass.Checked = False
chkPassNeverExpires.Checked = False
chkDisabled.Checked = False
chkLockedOut.Checked = False
chkRASUser.Checked = False
```

```
End Sub
```

```
End Class
```

### **Final Results/Recommendations**

This applet may be improved upon in the future by adding a text window that will display the commands to be executed as they are built. This would allow the user to copy the command string for use in a script. Another desirable feature would be the ability to build and run batch jobs given a list of data such as usernames, machine names, or resources.

**Project:** cvsfe

**Author:** Gunter Wambaugh

---

### Problem Background

CVS is a version control system that, among other things, tracks file changes by retaining previous versions of the file in a module located in the CVS repository. A user checks out a module, makes the desired changes, and then commits the module back to the repository.

By default, CVS will allow multiple users to edit a file simultaneously. With this model, conflicts can arise. These conflicts are resolved using an algorithm built into CVS. This algorithm performs well when the conflicting files are text files. Conflicts with binary files, such as those produced by Microsoft Word and Rational Rose, are more difficult to resolve. In such a case, it is up to the conflicting users to merge the changes on their own.

To avoid conflicts, a method for strict locking is needed. CVS provides mechanisms for strict locking in the form of extra commands. The commands require a certain order. It is easy to confuse the order and to skip steps—especially for users that are new to CVS.

### Project Goals

Cvsfe wraps the necessary CVS commands to utilize the strict locking model. Cvsfe is written in C with public usability as the primary goal. As such, cvsfe includes error checking and utilizes return codes from CVS. Cvsfe makes no assumptions about users or the environment other than the assumption that CVS is installed on the host. Additional CVS options can be passed to CVS through cvsfe. A man page for cvsfe has been created for hosts that support man pages. Cvsfe also utilizes the GNU tools autoconf and automake to ease the build process and to be more portable.

### Source Code and Documentation

```

/*
 * cvs.h
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

```

```

#ifndef CVS_H
#define CVS_H

/* Global environment variables. */
char *CVSEEDITOR;
char *CVSROOT;
char *PWD;

int verify_file( char * );

int check_for_editors( char *, char * );

void display_editors( char *, char * );

void remove_directory( char * );

void checkout( char *, char * );

void commit( char *, char * );

void lock( char *, char * );

void unlock( char *, char * );

void edit( char *, char * );

void unedit( char *, char * );

void export( char *, char * );

void get_environment( );

#endif /* CVS_H */

/*
 * cvs.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

```

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <limits.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/types.h>
#include "get_length.h"
#include "cvs.h"

int verify_file( char *module_name )
{
    struct stat file_info;

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    if ( stat( module_name, &file_info ) == -1 )
        return 0;
    else
        return 1;
}

int check_for_editors( char *cvs_args, char *module_name )
{
    char command[PATH_MAX], temp_file_name[PATH_MAX];
    struct stat file_info;
    int status;

    /* temp_file_name: ".{module_name}.tmp" */
    if ( get_length( 3, ".", module_name, ".tmp" ) > ( PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( temp_file_name, "." );
    strcat( temp_file_name, module_name );
    strcat( temp_file_name, ".tmp" );

    /* command: "cvs {cvs_args} editors {module_name} > {temp_file_name}" */
    if ( get_length( 6, "cvs ", cvs_args, " editors ", module_name, " > ",
temp_file_name ) > ( PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " editors " );
    strcat( command, module_name );
    strcat( command, " > " );
    strcat( command, temp_file_name );
}

```

```

if ( chdir( PWD ) == -1 )
{
    fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
    exit( 1 );
}

/* Execute command. */
status = system( command );

if ( WEXITSTATUS( status ) == 1 )
{
    fprintf( stderr, "cvsfe: There was an error with 'cvs editors'.\n" );
    exit( 1 );
}

stat( temp_file_name, &file_info );

/* Remove the temporary file.*/
if ( remove( temp_file_name ) == -1 )
    fprintf( stderr, "cvsfe: Could not remove temporary file %s.\n",
temp_file_name );

if ( file_info.st_size != 0 )
    return 1;
else
    return 0;
}

void remove_directory( char *module_name )
{
    char command[PATH_MAX];

    /* command: "rm -fr {module_name}" */
    if ( get_length( 3, "rm -fr ", module_name, " &> /dev/null" ) > ( PATH_MAX
- 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "rm -fr " );
    strcat( command, module_name );
    strcat( command, " &> /dev/null" );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    system( command );
}

void checkout( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];

```

```

int status;

/* command: "cvs {cvs_args} checkout {module_name}" */
if ( get_length( 4, "cvs ", cvs_args, " checkout ", module_name ) > (
PATH_MAX - 1 ) )
{
    fprintf( stderr, "cvsfe: Buffer overflow.\n" );
    exit( 1 );
}
strcpy( command, "cvs " );
strcat( command, cvs_args );
strcat( command, " checkout " );
strcat( command, module_name );

if ( chdir( PWD ) == -1 )
{
    fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
    exit( 1 );
}

/* Execute command. */
status = system( command );

if ( WEXITSTATUS( status ) == 1 )
{
    fprintf( stderr, "cvsfe: There was an error with 'cvs checkout'.\n" );
    exit( 1 );
}
}

void commit( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} commit {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " commit ", module_name ) > (
PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " commit " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )

```

```

    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs commit'.\n" );
        exit( 1 );
    }
}

void lock( char * cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} admin -L {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " admin -L ", module_name ) > (
PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " admin -L " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )
    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs admin -L'.\n" );
        exit( 1 );
    }
}

void unlock( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} admin -U {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " admin -U ", module_name ) > (
PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " admin -U " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )

```

```

    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )
    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs admin -U'.\n" );
        exit( 1 );
    }
}

void edit( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} edit {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " edit ", module_name ) > ( PATH_MAX
- 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " edit " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )
    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs edit'.\n" );
        exit( 1 );
    }
}

void unedit( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} unedit {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " unedit ", module_name ) > (
PATH_MAX - 1 ) )
    {

```

```

    fprintf( stderr, "cvsfe: Buffer overflow.\n" );
    exit( 1 );
}
strcpy( command, "cvs " );
strcat( command, cvs_args );
strcat( command, " unedit " );
strcat( command, module_name );

if ( chdir( PWD ) == -1 )
{
    fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
    exit( 1 );
}

/* Execute command. */
status = system( command );

if ( WEXITSTATUS( status ) == 1 )
{
    fprintf( stderr, "cvsfe: There was an error with 'cvs unedit'.\n" );
    exit( 1 );
}
}

void display_editors( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} editors {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " editors ", module_name ) > (
PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " editors " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )
    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs editors'.\n" );
        exit( 1 );
    }
}
}

```

```

void export( char *cvs_args, char *module_name )
{
    char command[PATH_MAX];
    int status;

    /* command: "cvs {cvs_args} export -r HEAD" {module_name}" */
    if ( get_length( 4, "cvs ", cvs_args, " export -r HEAD ", module_name ) > (
PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "cvs " );
    strcat( command, cvs_args );
    strcat( command, " export -r HEAD " );
    strcat( command, module_name );

    if ( chdir( PWD ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    if ( chdir( module_name ) == -1 )
    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", module_name );
        exit( 1 );
    }

    /* Execute command. */
    status = system( command );

    if ( WEXITSTATUS( status ) == 1 )
    {
        fprintf( stderr, "cvsfe: There was an error with 'cvs export'.\n" );
        exit( 1 );
    }

    /* command: mv {module_name} {PWD}/.{module_name} &> /dev/null */
    if ( get_length( 7, "mv ", module_name, " ", PWD, "/.", module_name, " &>
/dev/null" ) > ( PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "mv " );
    strcat( command, module_name );
    strcat( command, " " );
    strcat( command, PWD );
    strcat( command, "/." );
    strcat( command, module_name );
    strcat( command, " &> /dev/null" );

    /* Execute command. */
    system( command );

    if ( chdir( PWD ) == -1 )

```

```

    {
        fprintf( stderr, "cvsfe: Couldn't chdir( ) to %s.\n", PWD );
        exit( 1 );
    }

    remove_directory( module_name );

    /* command: mv .{module_name} {module_name} &> /dev/null */
    if ( get_length( 5, "mv .", module_name, " ", module_name, " &> /dev/null"
) > ( PATH_MAX - 1 ) )
    {
        fprintf( stderr, "cvsfe: Buffer overflow.\n" );
        exit( 1 );
    }
    strcpy( command, "mv ." );
    strcat( command, module_name );
    strcat( command, " " );
    strcat( command, module_name );
    strcat( command, " &> /dev/null" );

    /* Execute command. */
    system( command );
}

void get_environment( )
{
    PWD = getenv( "PWD" );
    if ( PWD == NULL )
    {
        fprintf( stderr, "cvsfe: $PWD environment variable is not set.\n" );
        exit( 1 );
    }
    if ( PWD[strlen( PWD ) - 1] != '/' )
        strcat( PWD, "/" );
}

/*
 * ci.h
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

```

```

#ifndef CI_H
#define CI_H

void ci( char *cvs_args, char *module_name );

#endif /* CI_H */

/*
 * ci.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include <stdio.h>
#include <stdlib.h>
#include "cvs.h"
#include "ci.h"

void ci( char *cvs_args, char *module_name )
{
    char choice;

    /* Verify the existence of the module. */
    if ( !verify_file( module_name ) )
    {
        fprintf( stderr, "cvsfe: %s does not exist, or you do not have the
proper permissions.\n", module_name );
        exit( 1 );
    }

    printf( "cvsfe: Checking in %s\n", module_name );
    fflush( stdout );

    /* Unlock the module. */
    unlock( cvs_args, module_name );

    /* Commit the module. */
    commit( cvs_args, module_name );

    /* Mark the module as 'not being edited'. */

```

```

    unedit( cvs_args, module_name );

    printf( "cvsfe: Finished checking in\n" );

    printf( "cvsfe: Delete %s locally? (y/n)", module_name );

    choice = 'x';
    while ( ( choice != 'y' ) && ( choice != 'Y' ) && ( choice != 'n' ) && (
choice != 'N' ) )
        choice = getchar( );

    if ( ( choice == 'y' ) || ( choice == 'Y' ) )
        remove_directory( module_name );

    exit( 0 );
}

/*
 * co.h
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#ifdef CO_H
#define CO_H

void co( char *cvs_args, char *module_name );

#endif /* CO_H */

/*
 * co.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.

```

```

*
* cvsfe is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with gtk-splitter; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include <stdio.h>
#include <stdlib.h>
#include "cvs.h"
#include "co.h"

void co( char *cvs_args, char *module_name )
{
    int do_remove;

    if ( !verify_file( module_name ) )
        do_remove = 1;
    else
        do_remove = 0;

    printf( "cvsfe: Checking out %s for editing\n", module_name );
    fflush( stdout );

    /* Checkout the module. */
    checkout( cvs_args, module_name );

    /* See if the file is being edited. */
    if ( check_for_editors( cvs_args, module_name ) )
    {
        printf( "cvsfe: %s is currently being edited by someone else.\n",
module_name );
        fflush( stdout );
        display_editors( cvs_args, module_name );
        if ( do_remove )
            remove_directory( module_name );
        exit( 1 );
    }

    /* Lock the module. */
    lock( cvs_args, module_name );

    /* Mark the module for editing. */
    edit( cvs_args, module_name );

    exit( 0 );
}

/*
* ex.h
*
* Copyright 2003 Gunter Wambaugh
*

```

```

* This file is part of cvsfe.
*
* cvsfe is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* cvsfe is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with gtk-splitter; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#ifndef CR_H
#define CR_H

void ex( char *cvs_args, char *module_name );

#endif /* CR_H */

/*
* ex.c
*
* Copyright 2003 Gunter Wambaugh
*
* This file is part of cvsfe.
*
* cvsfe is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* cvsfe is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with gtk-splitter; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include <stdio.h>
#include <stdlib.h>
#include "cvs.h"
#include "ex.h"

void ex( char *cvs_args, char *module_name )
{
    printf( "cvsfe: Exporting %s\n", module_name );
    fflush( stdout );

    /* Checkout the module. */

```

```

checkout( cvs_args, module_name );

/* See if the file is being edited. */
if ( check_for_editors( cvs_args, module_name ) )
{
    printf( "cvsfe: Note that %s is currently checked out for editing by
someone else.\n", module_name );
    fflush( stdout );
    display_editors( cvs_args, module_name );
}

export( cvs_args, module_name );

exit( 0 );
}

/*
 * rl.h
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#ifndef RL_H
#define RL_H

void rl( char *cvs_args, char *module_name );

#endif /* RL_H */

/*
 * rl.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *

```

```

* cvsfe is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with gtk-splitter; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include <stdio.h>
#include <stdlib.h>
#include "cvs.h"

void rl( char *cvs_args, char *module_name )
{
    /* Verify the existence of the module. */
    if ( !verify_file( module_name ) )
    {
        fprintf( stderr, "cvsfe: %s does not exist, or you do not have the
proper permissions.\n", module_name );
        exit( 1 );
    }

    printf( "cvsfe: Releasing %s\n", module_name );
    fflush( stdout );

    /* Unlock the module. */
    unlock( cvs_args, module_name );

    /* Mark the module as 'not being edited'. */
    unedit( cvs_args, module_name );

    /* Remove the module. */
    remove_directory( module_name );

    exit( 0 );
}

/*
* get_length.h
*
* Copyright 2003 Gunter Wambaugh
*
* This file is part of cvsfe.
*
* cvsfe is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* cvsfe is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License

```

```

* along with gtk-splitter; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#ifndef GET_LENGTH_H
#define GET_LENGTH_H

unsigned int get_length( int number_of_args, ... );

#endif /* GET_LENGTH_H */

/*
 * get_length.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include <stdarg.h>
#include <string.h>

unsigned int get_length( int number_of_args, ... )
{
    char *arg;
    int index;
    va_list argptr;
    unsigned int total_length;

    /* Initialize argptr. */
    va_start( argptr, number_of_args );

    total_length = 0;

    /* Sum the string sizes. */
    for ( index = 0; index != number_of_args; index++ )
    {
        arg = va_arg( argptr, char * );

        total_length += strlen( arg );
    }

    va_end( argptr );
}

```

```

    return total_length;
}

/*
 * cvsfe.c
 *
 * Copyright 2003 Gunter Wambaugh
 *
 * This file is part of cvsfe.
 *
 * cvsfe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * cvsfe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with gtk-splitter; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "cvs.h"
#include "co.h"
#include "ci.h"
#include "rl.h"
#include "ex.h"

void print_usage( );

int main( int argc, char *argv[] )
{
    char *cvs_args;
    char *module;
    char *command;

    /* Check command-line arguments. */
    if ( ( argc != 3 ) && ( argc != 4 ) )
    {
        print_usage( );
        exit( 1 );
    }

    if ( argc == 3 )
    {
        cvs_args = malloc( sizeof( char ) );
        cvs_args[0] = ' ';
        command = argv[1];
        module = argv[2];
    }
}

```

```

else
{
    cvs_args = argv[1];
    command = argv[2];
    module = argv[3];
}

/* Remove a trailing '/'. (Common error when using tab completion.) */
if ( module[strlen( module ) - 1] == '/' )
    module[strlen( module ) - 1] = '\0';

/* Set the environment variables. */
get_environment( );

/* Execute the proper command. */
if ( strcmp( command, "co" ) == 0 )
    co( cvs_args, module );

if ( strcmp( command, "ci" ) == 0 )
    ci( cvs_args, module );

if ( strcmp( command, "rl" ) == 0 )
    rl( cvs_args, module );

if ( strcmp( command, "ex" ) == 0 )
    ex( cvs_args, module );

/* If we made it here, neither of the commands above were specified. */
print_usage( );
exit( 0 );
}

void print_usage( )
{
    printf( "%s-%s Gunter Wambaugh\n", PACKAGE, VERSION );
    printf( "Usage: cvsfe \"[cvs-options]\" command module\n\n" );
    printf( "\tco\tCheck out module\n" );
    printf( "\tci\tCheck in module\n" );
    printf( "\trl\tRelease module\n" );
    printf( "\tex\tExport module\n" );
    printf( "\nNote that every cvs-option should be inside ONE pair of double
quotes.\n" );
    fflush( stdout );
}

```

The cvsfe "man" page is included below:

NAME

cvsfe - CVS Front End for reserved checkouts

SYNOPSIS

cvsfe [cvs-options] command module

DESCRIPTION

cvsfe is a front end to the concurrent versions system (CVS). It is intended to simplify the use of CVS with the reserved checkout model. CVS provides mechanisms for reserved checkouts, but several commands are involved. cvsfe wraps these commands into an intuitive command set.

ESSENTIAL COMMANDS

cvsfe co

Check out module\_name and set locks. The module is placed into the current working directory.

cvsfe ci

Check in module\_name and release locks. This command must be executed in the top level directory containing the module.

cvsfe rl

Release module\_name that was previously checked out and locked. No changes will be committed to the repository and the local files will be deleted. This command must be executed in the top level directory containing the module.

cvsfe ex

Export module\_name from the repository. Similar to cvsfe co, this command will check out module\_name; however, no locks are set and no cvs administrative directories are created. The module is placed into the current working directory.

OPTIONS

cvsfe accepts options to be passed to cvs. These options are ignored by cvsfe. See cvs\_options in the cvs documentation for details.

BUGS

When working with a remote repository, a user may be required to enter their password several times. This is because cvsfe issues several cvs commands, and each cvs command will likely require a password when dealing with remote repositories. Also, all cvs\_options must be placed into a single pair of double quotes.

AUTHOR

Gunter Wambaugh <techgunter@yahoo.com>

SEE ALSO

cvs(1)

### Final Results/Recommendations

Cvsfe is a working product. It is mainly for use on machines where the CVS repository is local. It can be used with remote repositories, but the user may be required to enter his password several times.

### References

Concurrent Version System. CollabNet. 30 April 2003 <<http://www.cvshome.org/>>

**Project:** etsumail2passwd.bash

**Author:** Robert Nielsen

---

### Problem Background

Currently, accounts for the Linux server, Einstein, must be generated by manually editing the input file used by the command "newusers". An E-Mail generated by the campus is used as a starting point for the file, but there are a number of changes which must be made before this file is of use.

### Project Goals

This script is designed to take the web-generated, class roll E-Mail and convert it directly to the form required by newusers. It requires that the E-Mail has been saved as a regular Linux text file. It will allow for student accounts to be generated with little overhead and it will also reduce the chance of errors related to account generation.

### Source Code and Documentation

```
#!/bin/bash

# Script name:   etsumail2passwd.bash
# Written by:   Robert A. Nielsen
# Date:        02/15/03
# Last modified: -----
#
# This script is designed to take the web generated, class roll E-Mail and
# convert it directly to the form required by newusers. It requires that the
# E-Mail has been saved as a regular Linux text file.

# Prompt the user for the name of the text file to read. Check to see if the
# file exists. If it does not, then re-prompt the user for the file name.
# NOTE - this script makes no attempt to validate the actual contents of the
# input file. If it has been modified from the original form of the E-Mail,
# the file produced may not be functional.

GroupID=507
PathToHome=/home/
PathToShell=/bin/bash
clear
echo "Please enter the input file name. Note that this script will overwrite
files with"
echo "the file name you enter and the extension .temp1, .temp2, and .out if
they exist."
echo "The .out file is the final output ready for use with newusers."
echo
validsource=no
while [ $validsource = no ]
do
    read filename
    if [ -e $filename ]; then
        validsource=yes
    else
        echo "Please enter the filename exactly as it was saved."
    fi
fi
```

```

done

# The script uses two temporary files named after the input filename.
cat /dev/null > $filename.temp1
cat /dev/null > $filename.temp2

# Since all E-Mail addresses contain "imail" (even faculty as produced by web
# interface), and "imail" should not appear in the mail headers, we'll use it
# to isolate the lines containing user information from the header lines.
grep imail $filename >> $filename.temp1

# The source E-Mail uses spaces instead of tabs to create columns, therefore
# we cannot use tabs as delimiters to locate the fields in the input file.
# Since names are of a variable length, we can't use the number of spaces to
# locate our fields. Since the name field contains spaces, we can't use
# single spaces as a delimiter. The "cut" command is unable to use multiple
# characters as a delimiter, otherwise this would be easier. So, to deal
# with this situation, we start by changing each instance of two spaces to a
# semi-colon. This allows us to deal with the spaces in the name fields.
awk -F" " '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";" $7 ";" $8 ";" $9
;" $10 ";" $11 ";" $12 ";" $13 ";" $14 ";" $15 ";" $16 ";" $17 ";" $18}'
$filename.temp1 > $filename.temp2

# Once we've established that a semi-colon will be the delimiter, we need to
# eliminate multiple instances so that each of our data items will be in the
# appropriate field. This first requires that we deal with the situation of a
# single space having been left by the conversion above. To do so, we swap
# any occurrence of ";" with a single semi-colon. Once we are certain that
# our fields have only semi-colons between them, we eliminate any instances of
# multiple semi-colons created by our earlier conversion.
awk -F";" '{print $1 ";" $2}' $filename.temp2 > $filename.temp1
awk -F";";";";";" '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";"}'
$filename.temp1 > $filename.temp2
awk -F";";";";";" '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";"}'
$filename.temp2 > $filename.temp1
awk -F";";";";" '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";"}'
$filename.temp1 > $filename.temp2
awk -F";";" '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";"}'
$filename.temp2 > $filename.temp1
awk -F";";" '{print $1 ";" $2 ";" $3 ";" $4 ";" $5 ";" $6 ";"}' $filename.temp1
> $filename.temp2

# At this point we isolate the username portion of the E-Mail address.
awk -F"@imail.etsu.edu" '{print $1 $2}' $filename.temp2 > $filename.temp1

# Now that we have our fields in the appropriate places, we want to change
# from our complete student ID number including dashes to only the last four
# digits. We'll use this as the initial password. We also need to account
# for the possibility of names that contain a hyphen. The following allows
# for a name to have up to four hyphens. The easiest way to eliminate the
# unneeded parts of the student ID number is to simply rewrite our line
# without the information that comes before the first two hyphens.
awk -F"--" '{print $3 "--" $4 "--" $5 "--" $6 "--}' $filename.temp1 >
$filename.temp2

cat /dev/null > $filename.temp1

```

```

# At this point, we've stripped out the unnecessary information and only have
# the last four digits of the student ID number, the student's name, and
# their E-mail username. We need to break this information into the
# appropriate fields and reorder them as needed by newusers. We also want to
# make sure that all usernames are in lowercase.
cat $filename.temp2 |
  while read studentinfo
  do
    studentid=`echo $studentinfo | cut -f1 -d";"`
    studentname=`echo $studentinfo | cut -f2 -d";"`
    usermail=`echo $studentinfo | cut -f3 -d";"`
    username=`echo $usermail | tr A-Z a-z`
    echo
    $username":"$studentid":":"$GroupID":"$studentname":"$PathToHome$username":"$Pa
thToShell >> $filename.temp1
    echo "Processing user" $username
  done

mv $filename.temp1 $filename.out
rm -f $filename.temp1
rm -f $filename.temp2
echo "Processing completed. See" $filename.out "for user file."

```

### Final Results/Recommendations

The user should see a file generated based on the name they selected. This file should be in the form of a “passwd” file for Linux. The command “newusers” will accept this as input and make the changes to the original “passwd” file automatically.

**Project:** Symantec Ghost AutoInstall for Configuration Management

**Author:** Trey Buck

---

### **Problem Background**

The computer science department at ETSU provides students with a “working lab” in which they can experiment with various software packages and software configurations. These projects often consist of many hours of work, consisting of several different work sessions. Since there are many more students than computers in the lab, the computers must be shared.

There is a risk that the work performed during one session may be destroyed intentionally or inadvertently by another student who needs to use the same piece of hardware. Often, students find themselves spending most of their time getting the system back to the same state it was when they last left it. This results in many hours of lost work.

What is desperately needed is a configuration management solution available for student use. The solution must be simple and effective. The higher the overhead, in terms of both complexity and time, the less useful it will be to the students. Here we examine the possibility of adapting Symantec Ghost AutoInstall to provide this functionality for Windows systems.

### **Project Goals**

Symantec Ghost is a package which allows administrators to effectively manage software in a distributed environment. Its primary function is to replicate the software installed on a prototype machine to other machines with similar hardware configurations. It does this by capturing a sector-based system image and providing the tools to deploy it to other machines.

AutoInstall (AI) is one of the components of the Ghost system. Its primary use is to provide software updating and patching capabilities for the Ghost system. AutoInstall achieves this by taking a “snapshot” of the prototype system. This snapshot contains information about the file system structure, registry entries, and other information.

After a snapshot has been taken, the package, patch, or update is applied to the prototype system and another snapshot is taken. AI then looks at the differences between the initial snapshot and the current system state. Any differences between the two are saved to a configuration file which AI Builder can use to create a single executable file. This executable file contains all the information needed to replicate the changes made on the prototype system to any other system whose software image was derived from the prototype.

The process used by AutoInstall can be summarized as follows:

1. Record the initial state of the system.
2. Monitor the installation of packages, patches, or updates
3. Compare the current system state to the original system state.
4. Record any differences in a configuration file.
5. Create an executable file that contains all the necessary information to replicate the changes.

Since AI is intended to enhance the Ghost system, it works very closely with Ghost. However, AI can also be used alone. The system is very easy to use and may provide the configuration management functionality needed in the lab.

### **Project Details**

An evaluation copy of Symantec Ghost 7.5 Enterprise Edition was obtained from the Symantec website. After the package was uncompressed, the AI component was installed. AI consists of two programs: AI Snapshot and AI Builder. AI Snapshot is responsible for taking snapshots of the system state, while AI Builder is responsible for building the executable file which will apply the changes to other systems.

Two tests were conducted to determine the AI's suitability for our solution. The first test began by taking a snapshot of the initial system state using AI Snapshot. After the snapshot had been taken, the AI Snapshot program was kept alive so that it could monitor the installation of our software.

The IPSwitch IMail mail server was chosen as the test package since it is a non-trivial application. It generates large amounts of registry entries and changes to the file system structure. It also provides a number of different services such as finger, IMAP, POP, SMTP, and many others. For these reasons, it was considered to be a good test candidate.

IMail was installed and configured. A unique selection of services was enabled in the IMail system as a fingerprint to judge how accurately the AI would be able to replicate the changes to the system. After the IMail configuration was completed, the still running AI Snapshot process was told to examine the system for any changes and create the configuration file. As expected, AI did just that.

Using AI Builder, the configuration file was used to build the executable. This also completed without incident. The IMail system was thoroughly removed from the system. The registry, file system, and MMC snap-ins were all examined to ensure no trace of the IMail system was left on the machine. The executable file was then run, which replicated the changes and rebooted the system. After the system rebooted, IMail was found to be installed, configured, and running exactly as it was before it was removed from the machine.

Now that we obviously have the capability to replicate software installation and configuration relatively painlessly, it would be beneficial eliminate the requirement that AI Snapshot be running while the changes to the system state are taking place.

After examining the documentation, it was revealed that the AI Snapshot application does not necessarily need to be running during the time the changes are made. It is recommended that the process be kept alive to guarantee the integrity of the initial snapshot data. This is discussed in more detail in the next section.

A second test was conducted which was much like the first. The exception was that after the initial snapshot was taken, the AI Snapshot process was killed. IMail was thoroughly cleaned from the system and another initial snapshot taken. Again, IMail was installed and configured

using a different selection of enabled services to serve as another unique fingerprint. After the IMail install completed, the system was rebooted, and the AI Snapshot application was run.

The first screen presented when AI Snapshot asks if a snapshot of the system should be taken. Ordinarily this is what is wanted, but not in our case. What we want is to use the snapshot taken before the installation of IMail as the basis for comparison. AI Snapshot provides exactly this option. When the option to use a previously taken snapshot is chosen, AI Snapshot will compare the current system state to the snapshot that was recorded previously.

After comparison, a second executable was built and IMail again removed from the machine. The machine was rebooted again and the executable file run. Indeed, the resulting system state was that we were hoping for: IMail installed, configured and running exactly as it was before it was removed

### **Lessons Learned/Problems**

Despite the success of the tests, there are some minor issues that may prevent the process from being successful under certain conditions.

This is a solution for windows platforms only. Though Ghost itself can capture and replicate Linux ext3fs filesystems, AutoInstall is a Windows application. Symantec does not currently have a Linux equivalent of AutoInstall, so this process is not applicable to Linux systems.

The Ghost manual recommends that AI Snapshot be kept running during the new software installation for a very good reason. If the rouge software installation were to change or overwrite the files that contain the initial system state information, the comparison between the initial and current system states would not be accurate. If the AI Snapshot process is alive during a new software install, it can lock those critical files to ensure their integrity. Killing AI Snapshot during the install removes this integrity guarantee. A possible solution to this problem is to backup the directory which contains the snapshot files and restore them prior to the compare stage. The files generated in our tests totaled just less than 20 megabytes for the IMail installation.

Another issue was encountered after running the AI created executable file to reinstall the IMail system. IMail was configured to use the Windows NT user database for authentication. After IMail was reinstalled from the AI image, IMail's connection to the NT user database needed to be re-established. Fortunately this was a simple operation, but it reveals a possible case where some tweaking may be required to return the system to the desired state.

One final limitation of the AI system is that only one snapshot may be present on the system at any given time. Some users may feel this is a limitation. It may be overcome by backing up and restoring snapshots as described above, or specifying changing the AI working directory as needed. AI does allow users to specify what directory it should be working in.

### **Final Results/Recommendations**

The AutoInstall component of the Symantec Ghost package appears to be a viable solution for providing configuration management in the computer science lab. Our test showed repeatability

and accuracy in the restoration of system states. The solution is effective and has little overhead. It is simple as can be expected, and takes a minimal amount of time. The system on which these tests were performed is a production machine with a large base of installed applications. There is 100 gigabytes of disk space on the system, of which 20 gigabytes are used. The scans for this system took a total of 20 minutes.

A useful addition to the system would be a mechanism to facilitate backing up and restoring the snapshot files. It would be difficult to make this mechanism work seamlessly as the working directory is user-configurable, thus, subject to change.

**References**

Symantec Corporation. 30 April 2003 <<http://www.symantec.com>>

**Product:** K12 Linux Terminal Server Project  
**Product Type:** Thin Client Computing Environment  
**Author:** Kao-Yee Chua

---

### **Problem Background**

Generally speaking K-12 public education often lacks enough budget to provide students with a good computing environment. This means that in schools students are learning computing skills on outdated computers for outdated software.

### **Product Placement**

The Linux K-12 Terminal Server Project (K12LTSP) is a free thin client Linux distribution that attempts to address this problem. Installing the K12LTSP and using a relatively powerful machine as a server along with several inexpensive older machines as clients, a school can have a computing environment containing recent and relevant software.

The K12LTSP is a project that is based on both the Linux Terminal Server Project (LTSP) and Red Hat Linux. The LTSP portion of the K12LTSP forms the thin client portion. The Red Hat portion forms the main bulk of the distribution with its wealth of packages and installation method. Several servers can be clustered.

### **Installation Overview**

Because the K12LTSP distribution is based on Red Hat's the installation is extremely easy. The latest version of the K12LTSP distribution is 3.0.1, and is based on Red Hat 8.0. The first step was setting up the simple infrastructure necessary for the thin client environment. This involved two machines (server and client) connected using a switch. The next step was to burn the provided ISO images to CD. The next step involved installing the distribution onto the server machine. During the installation options are given to install Workstation, Server, Custom, and the K12LTSP Server. The K12LTSP option was selected, and after a painless installation the server booted to an X Windows login screen with no problems onto a Toshiba Satellite laptop (Celeron 650 Megahertz system with 128 megs of RAM). Next the etherboot boot disk was created for the client. The distribution came with several images for several popular network card chipsets. The client machine (a basic Celeron 300 Megahertz system with 128megs of RAM) had a 3COM 595 chipset in it, and the K12LTSP distribution came with approximately eight images for this chipset. The fourth image that was tried worked, and the client machine booted up to the same X Windows login screen that was on the server. If the network card chipset image was not included in the K12LTSP distribution then getting the appropriate image would be easy. The site at <http://www.rom-o-matic.net/> allows users to dynamically generate images with various parameters. During the installation the only major difficulty was finding the correct boot image, but fortunately finding the correct one only took a few minutes. Overall the installation was extremely easy, and would generally not be difficult for any new users to Linux given relatively recent hardware.

The distribution allows administrators to customize clients to use different hardware and kernels by making changes to the `/etc/dhcpd.conf` file. When a clients first boots and contacts the server then it will see if an appropriate MAC address matches in this configuration file. Any

customizations for the client are specified, and sent via tftp to the client. The client computer I used did not need any customizations, and so no configurations were necessary. The main distribution comes with a generous set of applications including the OpenOffice productivity suite. The system is an almost standard Red Hat distribution, and so administration documents are readily available.

### **Lessons Learned/Problems**

The installation of K12LTSP went very smoothly, and no problems were encountered. The major lesson learned was that the distribution is an easy way to implement a low cost and yet powerful computing environment.

### **Final Results/Recommendations**

The community built around the K12LTSP is good for administrators new to Linux. The installation is very easy, and there are several links to documentation at the website. Also there is an active mailing list that allows administrators to ask for help. However the distribution can involve several difficult issues such as administering distributed file systems, configuring proxies and troubleshooting system performance issues. Thus it would be inadvisable for school systems to implement K12LTSP until the responsible staff member possessed adequate Linux administration skills.

The main web site also has links to Intel programs that provide free processors to eligible school systems. The list of school systems implementing the K12LTSP includes a majority taking advantage of Intel's generosity. In addition the site lists the hardware that several of the school systems possess, and many are using only systems that have dual Pentium II processors. This indicates that older versions can be run well on hardware that does not meet the current system requirements. Overall the K12LTSP is an excellent way for schools to implement a state of the art computing environment with relatively inexpensive hardware.

### **References**

McQuillan, James. Linux Terminal Server Project. 2003. 30 April 2003 <<http://www.ltsp.org>>  
K-12Linux Project. 30 April 2003 <<http://www.k12ltsp.org/>>

**Project:** Write a Scheduling Application for Computer Labs

**Author:** Adam Berry

---

### **Problem Background:**

The computer science department is responsible for staffing labs in Gilbreath and Wilson Wallis Hall. The customer, Robert Nielson, is in need an application that will automate some of the scheduling tasks. It was desired that the application be web-based.

### **Project Goals:**

Provide a GUI interface to schedule lab monitors to replace the current system of doing it manually.

### **Project Details:**

ETSU is mostly a Microsoft Shop, so the technology chosen for this web-based application was ASP (Active Server Pages). I first set out in search of an existing application that performed like I though I wanted the application to do. In talking with the client we identified a few requirements;

1. The system should be flexible for adding hours easily.
2. The system should display when the labs are open in week or month format.
3. The schedule would have little to no change once set.

With these in mind, I searched on ASP101.com and find some samples to work from. The first sample I downloaded was a simple calendar. From this, I attached a database and started customizing. I was pulling the date as a variable from the calendar script and generating a new record each time a date was clicked. This was counter-productive and I finally gave up, as this was not the way to go. The next script I found was a step in the right direction. It consisted of a calendar, which has the ability to add events. Those events carry a date, start\_time,end\_time, start\_date,end\_date, name, location, and description as attributes. This is very similar to what I needed but with a little extra. After eliminating a few fields and changing the database to reflect those changes, I had a basic working interface.

I can display a calendar for any month, and add unlimited events to each day. It is display in month format, so I have satisfied all requirements at this point. However, there were a few things I was not pleased with. For starters, when adding an event, a start\_date and end\_date must be entered. There aren't too many instances where the lab is open across days. When I tried to eliminate this requirement, no errors were thrown, and the database would add records, but they were not displayed on the calendar. I needed the end data to set the outputFilter for each week. For now, I will have to leave it as it is.

### **Lessons Learned**

There is a time to give up and I reached it with my first design. I am unhappy with the status of the second go-around. Even though it satisfies all requirements. I plan on taking it further.

**Products:** Fluke Network Inspector, Fluke Protocol Inspector, Deep Metrix IP Monitor, Sunrise Lan Explorer, Sunrise Traffic Max

**Product Type:** Network Monitoring Tools

**Author:** Todd Franklin

---

### **Problem Background**

The purpose of this project is to review several network monitoring tools to see which ones are the most useful without having frivolous features. This idea of what is the most useful may be a subjective statement so I will elaborate on this. What I am looking for in a network-monitoring tool is one that can monitor physical connections and the status of the different devices and also provide alerts when problems are discovered. I would also like the tool to include packet capturing, but this is not a must. The following is a review (or Bake-Off) of several network-monitoring tools. There will be a summary and score of the products at the end of this review.

### **Product Placement**

If a company is looking for network monitoring tools then one thing that is certain is that there is no limitation on what's available. The overabundance and varied claims of the companies that make these tools make it difficult to ascertain just which tool(s) would be right for any particular company.

### **Criteria**

The areas to look at for evaluating each product are simplicity, security, known algorithms, usability, functionality, good vendor (reliable, viable, trustworthy), integration, cost of ownership, and futures (is this the direction other companies are going?). The different types of network monitoring tools include device monitoring, traffic monitoring, tools used to specifically spy on employees, packet capturing tools (which can be used to spy on employees but is not specifically meant for that purpose), software and licensing tools, service monitoring tools, and variations and combinations of these. Some tools come in suites including many if not all of the above mentioned types of products. The products were chosen based on their price, advertised range of monitoring features, and advertised security features.

### **Installation Overview/Product Analyses**

The first tool I examined was Fluke's Network Inspector. Fluke was founded in 1948. They have long had a reputable name in testing and monitoring tools and are probably best known for their electronic monitoring and measuring devices. They have been thought of as number one in digital multi-meters, clamp meters, digital thermometers, and other tools for tradesmen for many years. As a licensed electrical contractor, I have always thought of their tools as the best on the market. Network monitoring is not listed as one of their primary industries, but they have a good reputation in this field as well.

Fluke Network Inspector monitors devices and services running on the devices. Additional tools can be added to integrated with it to make it a more versatile product. However, for the most part, this monitoring tool is mainly for use in monitoring the physical aspects of a network. It can be used as an aid in troubleshooting or in inventory of network devices. The installation process was simple and there were no complications. Running the agent had no apparent impact

on the performance of the network or the local host. The Fluke Network Inspector is primarily a tool for monitoring the devices on the network. The interface resembles is similar to Windows Explorer with a list of items in the left panel that can be expanded to show more specifics. The right panel displays particulars about any item clicked on in the left panel.

The devices were auto-detected. My network consists of a Windows 2000 Advanced Sever, a D-Link cable router, and two clients. All of the devices were detected. The router was listed as a server. When selecting the general item of devices in the left panel all four devices were visible in the right panel with a summary of each. The summary included name, IPX name (which wasn't applicable so it was blank), NetBios name, IP address, and MAC address. Double clicking or right clicking the device and selecting properties, brought up a properties box on the device. The properties box included tabs for Overview, Problems, Services, SNMP, Switch Inspector, and Notes.

The Overview tab displayed the NetBios name, Domain, a field labeled Microsoft (Windows 2000 Server was listed for the Windows 2000 server, Windows 95/98 was listed for the clients, and there was no listing for the router), and the MAC address. There were two boxes, one each for IP addressing information and IPX addressing information. The Problem tab had fields for description, detected, and last seen. The services tab listed services running on the device. The SNMP tab had fields for contact information and a community string which is the string used to query the device's SNMP agent. The Switch Inspector tab simply listed that no information was available and the Notes tab was a place to make notes about the device. All of the tabs had read only fields except for the Notes tab.

The left panel of the interface had a list of items under devices that included Fluke Tools, key devices (listed the server and the router), servers (listed the server and the router), routers (nothing listed), switches, managed hubs, printers (the printer on the network is not a network printer and was not listed), and hosts (listed the two clients).

There was also a local networks tree that, when expanded, included subnets, IPX networks and NetBios Domains. Under subnets was listed 192.168.0.0. Under NetBios names was listed "Athlon", the workgroup and under Athlon was listed the two Windows 98 clients.

Although the devices were listed under several categories, the same information was available from each location. This was the same as what was listed under devices which was covered above. On a large network this could be useful to find the specific device more quickly, instead of searching through the entire list of devices on the network.

The problems or potential problems to be monitored are established through the Network Inspector Agent console. There is a wide range of items to monitor, which are divided into three groups - errors, warnings, and changes. The following is a list of the items, which can be monitored.

Errors:

- key devices not responding
- incorrect subnet mask
- duplicate IP address

- duplicate NetBios name
- duplicate IPX network number
- duplicate IPX internal network number
- no Novelle file server found
- out of Novelle client logon licenses
- interface utilization exceeded 80%
- interface error rate exceeded 1%
- collision rate exceeded 20%

Warnings:

- incorrect IP address
- only device in IPX network
- only device in network using IPX encapsulation
- IPX service unreachable
- Novell client licenses about to overflow
- SNMP reported device rebooted
- interface utilization exceeded 50%
- collision rate exceeded 5%
- overlapped subnet mask

Changes:

- DNS name change
- IP address change
- IP service no longer seen on device
- IP service has resumed on device
- NetBios name change
- NetBios service no longer seen on device
- NetBios service has resumed on device
- device demoted to backup domain controller
- device promoted to primary domain controller
- device no longer seen on network
- Novell nearest file server has changed
- router interface has gone down
- router interface has come up

Any of these items can be selected for inclusion in a problem log or as the basis for notification. A different set of criteria can be selected for notification and another set to be logged. The methods available for notification are e-mail or pager. The problem log is listed at the top of the left panel and when it is selected it lists all of the entries in the problem log. By expanding the problem log, the specific areas of errors, warnings and changes are made visible and the respective items can also be accessed from here. There is no report for the problem log but there is an option to simply print the panel. There is no other option available other than selecting the general area of errors, warnings or changes to filter the problems recorded.

The system was tested using a variety of methods. The first of which was to simply disconnect the local server running the tool from the rest of the network. The results were not as expected. The error log reported that a key device, the cable router that was listed as a server, was not responding. It is very surprising that it did not report any other errors, warnings, or changes,

even though the settings were specified to report devices that were no longer seen on the network. This would certainly represent a problem, since if the server running the tool were disconnected by way of a bad NIC card or faulty cabling, the wrong information might be reported. A team of SA's could spend valuable time chasing down the wrong problem. There would however, probably be other indications that the local machine was off line, such as a list of all key devices not responding. One could only hope this was the case if they were using this product. In any event this response was less than adequate.

All of the other test that were performed went well. The IP address of one machine was changed and the change was promptly reported under changes in the problem log. The same host was removed from the network and it was not reported. This did result in a warning that the remaining client was the only device in the NetBios domain. When I reconfigured the router, a change was noted that the IP service was no longer seen on the device. Overall the performance was less than satisfactory.

Although the problem log was not available in a report, there were inventory reports and other more specific reports available - such as top interfaces by collision rate. The inventory reports, listed as IP Inventory and IPX Inventory, would certainly be of use in tracking devices on the network. The IP inventory report listed the router(as a server) and the services that it was running(DHCP, DNS, HTTP).

When the criteria that was mentioned earlier is applied to this monitoring tool we get a more complete picture. This tool would receive a good rating in the area of simplicity. It was easy to navigate after only a short time. The options were clear and easy to understand. Although additional tools are available they are installed separately and this could be done in stages to allow testing at each stage. Some tools came with the application but these were simple tools - ping, tracert, and Telnet that simply opened a command prompt window and launched the command.

In the area of security, the only indication that it is secure is the reputation of the company. The only indication of a possible security problem could be that this utility maps out all the devices on the network, including the services provided by each device, and has no longon or password control for the monitoring tool itself. Any security would have to be provided by the OS. Without any reports of problems from other customers there is little to go on. This is not an open source product. However, with scope of this tool this should not be a problem. Since it is not an open source tool there is also another area of security that will not be a problem - being easily hacked into because of well-known code.

The usability of this software rates well. It scanned the network and automatically reported on the devices without any configuration other than starting the agent. It is easy to use and the settings for reporting errors are easy to configure. This also leads into the question of functionality. This tool has some good features for monitoring network devices. Are these tools adequate? Yes for some companies but not so for others. No single piece of software will ever be the cure-all. However, there are additional tools available that integrate with the Network Inspector. One tool is the LAN Mapshot. It was installed but not able to run because Visio was not installed on the system. One review of Network Inspector noted that this was an impressive

aspect of the overall package. Another tool that was installed and did work was Fluke Protocol Inspector. This tool will be discussed shortly. If a company is looking for a device monitoring tool, this seems to have the right features. It could be more useful if it had a better way of tracking devices. For instance it could assign a unique identifier based on the MAC address so the device could be tracked throughout its lifetime on the network. It would also be an improvement if the tool would list any problems, present and past with the inventory reports.

This device seems to integrate well with a Windows system. As mention earlier there was no apparent impact on the resources of the network or the host it was running on. While using the Network Inspector, Windows 2000 performance monitors were running and there was no discernible difference in CPU usage and packets sent and received while the agent was updating. (Auto refresh frequency was 1 minute.) Initially, to learn about any IP, IPX or NetBios devices, the agent sends a small set of broadcast messages. It then listens for replies. It then sends protocol specific messages to determine if the responding devices are local to the agents sub-network. For each local device the agent sends a DNS query to get the devices DNS name. It also sends an SNMP query to get its SNMP group information. The agent also uses a passive discovery method by listening to unicast, broadcast and multicast traffic on the network. A central database can be created and registered. After doing this the database is ready to store the device information that is gathered by the monitoring tool. Upon initial start-up the user is prompted for the database and path. From everything indicated, the tool does not expect any network services that wouldn't already be running on the network. The Network Inspector is designed to support IP, IPX, and NetBios networks. It also requires Windows 98SE, ME, NT 4, 2000, or XP. In an all Linux environment this would require the installation of Windows machines on all subnets.

The cost for Fluke Network Inspector for a single console and up to 250-node license is around \$6,000. A single console license is available for \$2,000 and an upgrade of 50 nodes is available for \$695. Depending on the network the initial cost could be considerable. Ten subnets could run \$60,000. The training time should not be too high for a product such as this. This would probably be an initial investment of 30 hours to become competent with this tool. Once there are a few users familiar with the software it should not take very long to train new employees.

The “futures” for Fluke Network Inspector look encouraging. There are several good reviews out on the Internet, but what is more encouraging is the number of retailers that sell the product. One reviewer did mention the limitation that the product does not report on application layer issues. He also mentioned that the tool is limited to what the NIC can see and that some error may not be caught. Even so he thought it was worth the investment. Is the way companies are moving? It is hard to say, but Fluke also has many newer products and they seem to be looking to the future themselves.

Another Fluke product that was installed along with the Network Inspector software was Protocol Inspector. The Fluke Protocol Inspector monitoring tool is multifaceted tool for monitoring network traffic and capturing packets. It allows the use of packet filtering. It also has an assortment of bar charts that change to reflect the traffic on the network. These charts include a host table, application layer host table, application response time monitor, address mapping monitor, Vlans monitor, application layer matrix monitor, network layer matrix, host

matrix, network layer host table, protocol distribution monitor, frame size distribution, a utilizations/errors strip chart and a local NDIS 802.3 module (capture and monitor mode). Most of these charts list the top ten stations for the listed activity. These charts also have a table tab, which presents the information in numeric form, which would not be as flashy but would seem to be more useful. The protocol inspector has a broad range of alarms that can be set. These include:

Expert Alarms:

Application layer

ICMP - excessive ARP

Transport layer

TCP/IP retransmissions

Network layer

broadcast (of several types)

illegal Vlan ID

IP time to live expiring

illegal network layer source address

Ethernet

illegal MAC source address

excessive collisions

excessive broadcast

Token ring

similar to Ethernet (no excessive collisions)

Response time alarms: Based on the response times of various components(FTP, Telnet, SMTP, HTTP, DNS, POP, NFS and more).

Network layer alarms:

IP packets and octets

IPX packets and octets

ARP packets and octets

NetBios packets and octets

Ethernet alarms:

frames

broadcast frames

multicast frames

collisions

Token Ring: Same as Ethernet except for collisions.

All of these different types of alarms had fields, which could be set by the user. These fields included sample type (delta or absolute), rising value, falling value, severity (normal, mild, major, critical), actions (message, log file, e-mail, pager, stop and save, restart, auto save), interval, and enabled.

The Protocol Inspector has two basic modes of operation - monitor and capture. It can also be set to both monitor and capture packets at the same time. There was a significant increase in CPU usage when the program was in either mode. The tool produced a 8% to 12% increase while in active use.

The packet capture feature had several options that could be changed, such as buffer size and filters. The filter feature had templates that could be loaded or a custom template could be created.

After capturing packets, the capture view feature could be selected to view the captured packets. The version of Protocol Inspector that was being tested was an educational release and only allowed the viewing of the last 250 packets. The information available about the packets included an ID, status, elapsed time, size, destination, source, and summary. More detailed information about each packet could be obtained by selecting an individual packet. This information included such things as time of arrival, frame status (good frame), and information under the heading of the particular protocols associated with the frame. Under IP it showed type of service, flags, TTL, checksum, and IP source and destination address. Under TCP it listed source and destination ports, flags, sequence number, header length, and window size. There was information for UDP, HTTP and much more depending on the type of packet. The HTTP packets could actually provide the URL and sometimes it could provide some HTML code.

The usefulness of this type of tool could certainly vary with the knowledge of the users. The alarms could definitely be useful, while the colorful charts are probably not as effective as some would think. The packet capturing can tell a knowledgeable SA much about his network. The flow of traffic would be one tool to help quantify a networks status and also help plan for future growth. However, with only the raw data it would be an overwhelming job to translate it into information that could be of use. Some other tool would have to be used.

The Fluke Protocol Inspector is a tool that is meant to be interfaced with one of Fluke's hardware device analyzers. The Fluke protocol Inspector Pro cost just over \$6,000. It is an inline network-monitoring device, but has much more capabilities than just the software package. The software is not available by itself except as an educational version. It would take a substantial amount of time to research this product before an informed decision could be made on all aspects of its use and costs. However, the software itself seems impressive.

When evaluated in terms of simplicity this tool is far from simple. It has more charts for different types of evaluations than can be imagined. In addition, the interface itself has many settings that would take a while to comprehend. The amount of information gathered through the packet capturing is overwhelming to the point that it would take an additional tool to make it useful to its full potential. Although the tables are supposed to provide a good analysis of the data, its seems improbable. I am not aware of any reports that can be generated by the program besides just printing the current window.

While the alarms may provide additional security there is no basis for a determination about the security of the monitoring tool itself. Once again, the company is well respected, but it is hard to tell how much faith in the security of the system is justified based on this alone. This is not an open source tool, which may provide some relief from hackers.

The usability of this product might be high, but it is a complicated tool and would take some time to reach that conclusion. The configuration of the alarms is not too difficult to understand and the capture and monitor modes also do not present a problem. Once again, the complexity would lie in being able to use the information gathered to its full extent.

The product has good functionality in that it provides a lot of features and ones that could be useful with training. However, the colorful charts seem to have little real value and the program would be just as well if only the numeric tables were used.

Vendor issues have been addressed previously for this company. The only product specific vendor issues are the fact that this product cannot fully be tested without the purchase of expensive hardware. This also affects the integration of the product. The Fluke Protocol Inspector could not be integrated cheaply into a network. This definition once again depends on the size of the network. The product does put a noticeable load on the CPU of the local host. The monitoring tool is meant to interface on special hardware, but the OS needs to be Windows 9x, NT, 2000, or XP. The diagnostic hardware also will affect the cost of ownership. However, it is not significantly more than the Network Inspector. The real cost of ownership issue would be in training dollars. It would probably take at least a couple of weeks to become familiar with most of the features and it would be a continuing process. All of this is absent the added knowledge that would be needed to configure and maintain the in-line diagnostic hardware.

The “futures” of this product would seem to be good, if looking at the number of retailers who sell it means anything. Doing a search produced an interesting number of colleges who had lesson plans and other assignments based on the Protocol Inspector. However, this could be due to the fact that there is an educational version, which does not require the diagnostic hardware.

It is impossible to do a complete review on this monitoring tool without all the relevant hardware, but it certainly has potential.

The next monitoring tool that was examined was IP Monitor by Deep Metrix. This monitoring tool uses Internet Explorer (or Netscape Navigator) as its interface and is installed as a service on a Windows NT 4.0 (service pack 5), 2000, or XP machine. The browser interface requirement required that Internet Explorer be upgraded to the latest version. The initial configuration simply requires the IP address of the local host and setting a few permissions - whether to require a password to access this tool. Then networks can be added. After the network is added, the IP Monitor scans for devices using NetBios.

To use this tool, groups, monitors (specific profiles), alerts, and reports must be configured. The groups are used to organize the monitors together with other similar monitors. This is useful when creating alerts and reports. All alerts about a certain group of monitors can be sent to one person, while alerts from another group can be sent to a different person. When a monitor is created, it can only include one item to observe. There is a wide range of items that can be monitored. These include:

Application level and in depth testing:  
HTTP, POP3, SQL, DNS, FTP, and others.

Windows NT and file change(i.e.: Dr Watson Logs):

Service, ODBC/SQL, execute 3rd party monitor, event logs, file change, drive space monitor, Active Directory and others.

Frequently used TCP/IP monitors:

HTTP, SSL, PING, FTP, LDAP, SNMP, Kerberos 5, and others.

Other TCP/IP monitors:

IRC, Finger, SNPP, NTP, Gopher, Radius, and others.

In creating a monitor settings are available for frequency of testing, interval of time after which to stop testing (if there is no response), and interval of time after which to retry (after receiving errors). There are also settings to process alerts after a certain number of failures and to stop alerting after a certain number of notifications.

There is also a nice feature that allows the SA to input a maintenance time during which the monitored item will not be available, so that false errors and alerts are not generated. Alerts can be configured and be associated with all monitors, only a group of monitors, or only one monitor. The alerts can be processed through e-mail, instant messaging (winpopup/"netsend"), Alphanumeric pager, recovery script, or sending an SNMP trap. The last two options require third party software.

Reports can be generated. These can be an availability report, downtime report, response time report, health report, trouble report, or a diagnostic report. These reports can be based on recent activity, end-of-day, end-of-week, or end-of-month. There are options for including a historical trend analysis, detailed reporting, or creating a hyperlinked index. The reports can include one or more data sources (monitors).

Finally, there is another nice feature, which allows an SA to remotely start, stop, pause, and restart services on servers and workstations. This option is only available if the account under which IP Monitor runs has the necessary permissions.

This is really a very simple piece of software. The configuration is performed through a wizard-like interface (in a browser window). It is pretty easy to navigate after only a short while. The settings are easy to understand. Since only one item is chosen when each monitor is configured, the process remains simple. This would, however, result in more time creating multiple monitors. The idea of multiple monitors, makes delegating areas of responsibilities in response to alerts very simple. The security requires a username and password in addition to the system authentication. However, the tool uses HTTP to communicate which could cause some security concerns.

The usability follows from the simplicity of this tool. The configuration is simple and can be easily viewed and verified. This product also provides a wide variety of specific items, which can be monitored without, being too complicated. As mentioned earlier, individual monitors consist of one item to monitor.

The question of integration is mostly based on the fact that this like most similar products is really for a Windows-centric network. There could be some problems with the use of HTTP for communication but several proposals are provided in the help guide. It seems to put little or no load on the supporting systems. The reports can be stored centrally and easily moved if necessary.

The cost of ownership for IP Monitor would not seem to be very high. The initial cost is \$695 and the cost of training should not be as much as other monitoring tools. Within a week, a knowledgeable SA should have a good grasp of the fundamentals of this system - maybe sooner.

Deep Metrix was created in 1992 and has had several products out that have seemed to be widely accepted. The IP Monitor has been out for six years and has had very good reviews. This product has been touted as a product that excels in the ease of use. Considering the ease of use, the low cost, and the range of monitoring which is available, the “futures” for this product should look very good.

The last product looked at was a combo package of Sunrise Lan Explorer and Traffic Max. The Lan Explorer uses NetBios Lookup and DNS queries for name resolution. The documentation also mentions that it uses auto-discovery but does not explain exactly what the process is. After auto-detecting the devices on the network, an address book keeps a record of all devices using MAC addresses, IP addresses, and host names.

Lan Explorer is supposed to intercept all packets coming to or going from the local machine and intercept all packets on the network segment while in promiscuous mode. The interface consists of a task panel to the left with three buttons to select the type of panel. The traffic panel has icons for different tables and charts. These include a matrix table, host table, TCP/UDP port table, matrix chart, host chart, TCP/UDP port chart, and traffic monitor.

The host table simply shows the hosts that have been detected. They are listed by address (which may be an IP address or a NetBios name), octets ratio, total packets, total octets, octets and packets in and out, broadcasts, multicasts, first time stamp and last time stamp. Icons at the bottom of the window can toggle between the IP and MAC window and set the polling interval, and resort the listings according to the different fields in descending or ascending order. There is also an icon to set a filter, which can be based on a protocol or an address. The TCP/UDP port table has similar information available based on ports. The matrix chart is a pie chart showing the allocation of traffic on the listed machines. An icon allows this chart to be toggled to include or not include broadcasts. The host chart and port chart are pretty much the same thing. Each chart has the icons at the bottom of the screen to change some options. The only discernible difference is that the different charts have slightly different views. The charts are each based on variations of the above-mentioned fields. Clicking an icon can change this and various combinations can be created. All of the charts and graphs update automatically as information is gathered from the network.

The statistics panel gives more colorful graphs of statistics such as Telnet octet rate, HTTP octet rate, and so on and has the capacity to make custom charts, which may be the most useful tool in this application so far. The custom chart can be created to monitor any of the common ports and some not so common ones.

The last panel is the Alarm log. It gives any alarm information that has been generated by the monitoring tool. The packet capture trigger can be set to start or stop the packet capture under different conditions. Packets can also be captured manually. The capture will stop after the buffer is full, but this can be changed to wrap around buffer or better yet the buffer size can be changed. Once the buffer is full or after the capture has been stopped, the packets can be viewed. The information available is very much like what was available in Fluke's Protocol Inspector. Destination and source address, listed by IP address or NetBios name, protocol, summary, size, and tick (msec.) was listed for each packet. Clicking on the individual packet also gave almost the same information as Protocol Inspector. Ethernet, IP, ICMP, HTTP and other fields were available with details listed in each field. It would seem that the same packet would give the same information with either tool.

There are no reports with Lan Explorer. However, the charts can be printed. This tool may be designed more with presentations in mind than in depth monitoring. There is also no method of notification of alerts other than simply logging the alert.

The Sunrise Traffic Max was very similar with all the same charts and graphs. The fundamental difference was that Traffic Max did not have the option of capturing packets, but it did offer better alarm capabilities and notification. The notification included e-mail, pager, SNMP trap and logging. The areas for sending the alarms covered network security items, like failed authentication, and performance items like collision rate threshold and error rate threshold. The report option, which was totally absent in the Lan Explorer, allowed specific items to be reported on, and the schedule could be set daily, weekly or on a custom basis. The items that could be the criteria of a report were all of the items in the tables and charts, different packet rates, collision rates and other performance issues.

One item of note is that both of the Sunrise tools locked up the server on several occasions.

These two systems don't rate well in terms of simplicity, usability, functionality and integration. It would be hard to trust the vendors who put out a product, which was big on flash and small on real practical use. There may be a big need for a nice pie chart to take into a meeting to show the team the distribution of network traffic. However, there is probably a much better tool at a better price. The Lan Explorer cost \$910 and Traffic Max cost \$1100. It would be hard to justify this on any but a large network. The cost of ownership would be much higher with the training needed for a flawed product. If one of the Sunrise tools took down a server at a peak time of usage, then the cost of ownership could be much more.

### Final Results/Recommendations

The ratings of the products are found to be as follows:

	Vendor Issues	Security	Functionality	Total Cost
Fluke Network Inspector	4	3	4	4
Fluke Protocol Inspector	4	3	2	2
Deep Metrix IP Monitor	3	2	5	4
Sunrise Lan Explorer	3	3	2	2
Sunrise Traffic Max	3	3	2	2

Rankings are based on a scale of 1 to 5 with five being the highest ranking. Features and vendor issues rated together, simplicity, integration, and usability - all are rated under functionality. I found the Fluke Network Inspector and the Deep Metrix IP Monitor to both be very useful and functional tools and would recommend either of them. I prefer Deep Metrix IP Monitor overall. Even though it does not have the reputation that Fluke tools do, the functionality was far superior to all of the other tools.

**Product:** Microsoft Web Application Stress Tool

**Product Type:** Web Application Stress Tool

**Author:** Robert Nielsen

---

### **Problem Background**

From businesses large and small, to educational institutions, from government agencies, to individuals, from most any entity to most any other, web sites are becoming increasingly popular and often profitable offerings. When offering web service, be it to local users or users across the globe, one of the worst things that a site can face is poor performance. Thankfully, there are many tools available to assist administrators who want to increase the likelihood that their web server will perform up to their level of needs. One of the functions offered by many of these tools is the ability to generate test loads against a web server for the purpose of stress testing. Stress testing a web server allows the administrator to have an idea of how much activity a particular server can handle successfully by using a small number of systems to simulate a large number of users on many systems. This type of testing can produce knowledge that assists sites that are already running in determining if hardware upgrades are needed and, combined with estimates on anticipated use patterns, can help ensure that new sites can provide the capacity needed.

### **Product Placement**

To investigate the capabilities of stress testing software, it was decided to attempt to measure the capacity of one of the Computer Science Department's web sites. In particular, the new helpdesk site was chosen as the target of the testing. Since the side effects of stress testing can include temporary loss of functionality by the site being tested, the decision was made to pull the content of this web off to another web server. This server has only a fraction of the capacity of the actual Department server, so the expectation is that if this box can handle the load, the production server should have no problems handling the same load. The server used for testing was a Dell GX110 with a 667 MHz PIII processor and 512 MB of RAM. The web server implemented was Microsoft's Internet Information Server, the same server software running the Department helpdesk. This server had no other service or software running at the time of testing, other than those being tested and the Microsoft Performance Monitor. Microsoft Performance Monitor was used to provide additional data to assist in determining where exactly the system encountered problems during the testing.

### **Installation Overview**

The first attempts to test this server were carried out by using Hammerhead 2, an independently produced product available under the GNU Public License. This product is designed primarily for use under FreeBSD, though it is also supposed to be supported by most version of Linux. The software installed easily, though it did spread portions of itself through a number of directories. Configuration for the first attempt was left mostly at the defaults, with only the destination modified. Running the test at this level seemed to work normally, with the web server logs indicating a pair of accesses from the client. As the next step in testing, an attempt was made to increase the number of clients that were being simulated. With the configuration file updated, a second round of testing was attempted. Unfortunately this testing did not go very well. Though several different configurations were tried, the Hammerhead was never able to

generate more than a single pair of access to the web server. The configuration changes attempted included modifying the run time, the number of sessions, enabling IP aliasing, etc. After several iterations of modifying the configuration and running Hammerhead, it was deemed best to start investigating other software packages for stress testing.

A web search produced a document dedicated to testing IIS with freely available tools. One of the tools mentioned in this document was the Microsoft Web Application Stress Tool. This tool is freely available from Microsoft and though missing some of the more interesting features of Hammerhead it appears to be a much more mature product. Some of the features available in WAS include the following:

- Multiple methods for creating access scripts including manual creation, creation by browser recording, creation by use of previous IIS logs, and creation by pointing WAS at the content tree.
- Multiple client interfaces including a C++ client, an ASP client, and the ability to create new clients.
- The ability to create multiple users and to include passwords for validation.
- The ability to validate users by multiple methods including DPA, NTLM, and SSL.
- The ability to handle cookies for the Web Application Stress users.
- The ability to run as a service under NT and Windows 2000.
- The ability to use multiple machines for testing.
- The ability to control multiple test clients from a central location.
- The ability to simulate modem use via bandwidth throttling.
- A query-string editor that allows you to create and save templates for use with multiple tests.
- The ability to generate summary reports of the measured performance data.
- DNS support for client lookup and for cluster testing.
- The ability to create page groups so that you can logically group files for access and control script flow.
- The ability to customize headers so that you can simulate access by most browsers.
- The ability to set delay times between requests, so that you can produce exact time sequences and test for race conditions.

### **Lessons Learned/Problems**

Running the test at this level seemed to work normally, with the web server logs indicating a pair of accesses from the client. As the next step in testing, an attempt was made to increase the number of clients that were being simulated. With the configuration file updated, a second round of testing was attempted. Unfortunately this testing did not go very well. Though several different configurations were tried, the Hammerhead was never able to generate more than a single pair of access to the web server. The configuration changes attempted included modifying the run time, the number of sessions, enabling IP aliasing, etc.

### **Final Results/Recommendations**

The testing attempted includes only a small subset of the capabilities offered by WAS, but was enough to give an idea of what both the software and the server were capable of doing. As with Hammerhead, the testing was started by using the sample configuration files. A single session was able to produce 3428 hits on the server over a two minute period. That works out to an

average of 28.57 requests per second. The number of sessions was increased from one to twenty which resulted in 27928 hits or 232.73 requests per second. With the server responding well so far, the number of sessions was increased to forty. This resulted in only 27750 hits, a decrease of about 175 from what we achieved with 20 sessions. Reducing the number of sessions back to 30 resulted in 28190 hits or 234.92 requests per second. This would make it appear as though our maximum number of requests per second for our test server, using the sample files, is likely about 235.

With this basic testing completed, a script was generated based on the actual files which comprise our helpdesk web. This was done by use of the web browser recording option. With the script in place, a total of sixty tests were run. This testing started at a single session and continued to 2500 sessions. The following chart is a summary of some of the more informative tests.

<u>Sessions</u>	<u>CPU Range</u>	<u>CPU Average</u>	<u>Requests/Sec</u>	<u>Socket Errors</u>
1	10-40%	20%	1.84	<10
2	20-60%	40%	3.52	<10
4	40-80%	65%	5.95	<10
8	50-80%	75%	6.83	<10
10	65-85%	77%	7.00	<10
20	65-90%	78%	7.21	<10
40	65-98%	79%	7.51	<10
50	70-99%, then drops to 55-80%	80% - values past here are invalid	7.85	<10
100	NA	NA	8.17	<10
190	NA	NA	9.50	<10
200	NA	NA	10.00	<10
210	NA	NA	9.90	<10
1000	NA	NA	5.12	<10
1975	NA	NA	3.51	<10
2000	NA	NA	2.87	2971
2500	NA	NA	2.62	105020

The maximum value of requests per second, 10.00, is achieved at 200 sessions. In many cases this would indicate that our maximum capacity was 10 requests per second. When the log files of the server were analyzed, it was determined that the server began to regularly lose the ability to properly process ASP at the fifty session level though. It is due to this that the CPU values for tests beyond 50 sessions were dismissed. Our finding that the system became unstable at about 80% CPU utilization corresponds with the tools help files which indicate that problems are likely to arise when the processors are running at or above 80-85%.

From these tests it would appear that the Department servers should be safe at use below 10 sessions per second. This should be far above the level that the Department would actually require, though accesses to other websites on the same server could radically affect performance of the helpdesk site. It was interesting to note how much impact the content of the pages had on

performance. The sample files had a maximum requests per second that was about 25 times what could be achieved with our actual pages. This seems to make it clear that testing of the actual pages, instead of sample pages, should be done if at all possible when performing stress testing.

**References**

"MS Web Application Stress Tool". Microsoft Corporation. 2003. 29 April 2003

<<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/intranet/downloads/webstres.asp>>

Wong, Geoff. December 2000. 29 April 2003 <<http://hammerhead.sourceforge.net>>

**Product:** Webmin

**Product Type:** Utility

**Author:** David Frazier

---

### **Problem Background**

Using a default installation of Red Hat, there are no good tools for graphical remote administration. I have found a tool known as Webmin that seeks to address this need. Webmin runs via HTTP on port 10000 of the machine it is installed on. It allows any SA with the appropriate login and password to do server administration tasks in a Web environment. All that is needed at the remote site is a Web browser and a connection to the Internet. Webmin is available free of charge at [www.webmin.com](http://www.webmin.com).

### **Product Placement**

Let's say that a business has some Linux servers installed that run services such as Email, Web and ftp. In addition, the servers allow programmers to log in and do development work. If these servers are running a graphical Linux version such as Red Hat 8.0, administering the systems while at the console is straightforward. There are graphical front ends to allow most of the configuration work to be done. The senior SA's will almost always opt for the command line way to accomplish tasks such as setting up new users, and restarting services. Junior SA's will most likely feel more comfortable with the new, graphical way. As long as the SA is directly connected to the server, they can choose whichever method they prefer.

There will be times when it will be necessary to administer the server from a remote location. This remote location could simply be from the SA's desk instead of in the machine room, or it could be from a remote site. As long as telnet support is available, the Senior SA will have no trouble logging in and taking care of things via the command line interface. A junior SA, comfortable only with the graphical way of doing things, will be a distinct disadvantage. Documentation could go a long way towards bridging this gap, but there are no guarantees that the documentation will be available when it is needed. A graphical user interface can also cut down on errors as it allows choices to be made with less chance for a typo.

What tasks can Webmin do, and does it do them well? It is a realistic choice for junior SA's who need a helping hand? Does it offer any benefits to Senior SA's? Are there any security concerns with the use of this product? These are the questions that I will explore in this report.

Webmin describes itself as a "web-based interface for system administration for Unix". It is available from Caldera under the BSD license. The design goal for Webmin was to combine administration tools together under one product, and make them work within a Web browser using Perl CGI scripts. Webmin allows SA to view and modify most if not all services and actions needed to administer a server. Webmin is not supposed to make unnecessary changes to system files, or make any changes at all if it does not understand what you are trying to do. Webmin does not alter the format of any configuration files, so it can be used in addition to command line techniques. It attempts to provide a simple, consistent interface into the services it provides.

## Installation Overview

Installation of Webmin was very easy. I opted to use the RPM, which installed itself with no intervention from me. A tarball is also available for a variety of Linux and Unix OS's. There was no configuration necessary. My Apache server was already up and running, so I simply opened Mozilla and pointed to <http://localhost:10000>.

Webmin asks for a username and password. Initially, you must use the root username and password to gain access. Once you are logged in, Webmin presents you with seven tabs that contain the tasks you can accomplish. The tabs are Webmin, System, Servers, Networking, Hardware, Cluster and Others. Again, no configuration is needed. Webmin discovers your configuration automatically.

By default, the Webmin tab is open. The most useful tool in this section is the Webmin Configuration. In this section there are tools that allow you to configure Webmin and add some additional security. For example, the IP Access Control tool allows you to define a list of trusted IP addresses that are the only ones that can access Webmin. In this section you can also change the default port that Webmin listens on. There is also a tool to specify what logs are to be kept. You can also change the User interface, and even add themes. For additional security and flexibility, you can add Webmin users, and determine the levels of access each has. This could work extremely well in our example. If there are junior SA's who can be trusted to perform some tasks, but not others, you can simply tell Webmin which tools they have access to in their user profile.

The second Tab available in Webmin is the System Tab. This section allows you to make changes to the configuration of the server. There is a Bootstrap/Shutdown tool that allows you to edit the startup and shutdown configuration files. For example, on my Red Hat server this would be the information found in the `/etc/rc.d/inti.d` file. This tool allows you to check off which services should be started upon boot. There are also buttons to actually Reboot or Shutdown the server.

The Disk and Network Filesystems tool gives you a detailed table of all of the filesystems listed in `etc/fstab`. From this tool you can mount and unmount filesystems in addition to setting the configuration options for each existing filesystem. This tool is very useful, as you can visually see all the filesystems that are currently mounted. You can also choose to allow quotas on each filesystem. This tool could be especially useful for our Junior SA's, as they do not need to know which file systems are mounted, or how to make changes. If they know what they need to do, they can accomplish the task visually.

Other tools in this section allow for the user to view currently running processes. If you click on an individual process ID, Webmin will provide more information. This includes such useful information as the command that created the process, the CPU usage and the niceness level. You can also determine which files each process is using.

You can also find tools in this section to allow you to schedule jobs and create Cron jobs. Webmin allows even the most novice SA to easily create jobs that will execute at a certain time, or Cron jobs that will execute again and again. Cron jobs are often found in different places in

different OS's, and can be difficult to find and create. Webmin makes Cron jobs easy to create. It had been a while since I had created a Cron job, and that was on HP-UX. With Webmin, I easily created a Cron job to run Tripwire every morning.

One of the most useful tools in Webmin is the Software Packages Manager. I found this tool to be useful even on a stand-alone machine. The Software Packages Manger allows you to view all of the packages that are currently available on the server. This service is supported on all major versions of Unix and Linux, except Irix. Each package can have its properties viewed and changed. You can also use this tool to install new software, or upgrade existing software. I found this feature extremely easy to use. It is very useful to be able to see what software is installed, and upgrade it from a central location, especially if you can do it remotely.

This section also includes access to all system logs. System logs are often scattered all over the file system. With this tool they can all be viewed in one location, in a consistent manner. This is another tool that I found very useful on a stand-alone system. It would be even more useful on a remote system. Junior SA's could find the log they need without having to know what directory it is kept in.

The Users and Groups tool looks very similar to the Red Hat graphical Users and Groups tool. The advantage to this tool is that you can do it remotely. When you run the tool, you are presented with a list of users. It was instructive to me to see this list on my stand-alone machine, as there were some users that I had forgotten that I created. You can also get a list of groups and who belongs. Clicking on a user's name displays more detailed information about the user. This information can be viewed or updated. It is extremely easy to add new users. I added a couple and found that the visual prompts forced me to make informed decisions instead of just relying on me remembering which options I needed to set. The new group entry was also extremely easy. I could just pick existing users from a list to populate the group. If a Junior SA were given the task of User and Group account maintenance, this would make the job a snap, and would be accessible from anywhere there was a browser and a Internet connection.

The next tab is the Servers tab. The tools in this section allow you to display and configure services and daemons. Each service or daemon is automatically detected, can be selected. There were many services available on the machine I was testing. I decided to explore the Apache service.

The Apache tool allows for various options to be set. You can set information about which kinds of processes to accept. I would have no idea how to configure this information manually. In addition you can add Networking information such as which port to listen on, as well as set up virtual servers.

Another nice tool allows you to add Apache modules from a simple checkbox. All common modules are listed, and you simply check-off the ones that you want. I needed to add php support for my Apache server, so I simply checked on the mod\_php4 box. After a restart of the server, it worked perfectly. There is also a graphical tool to add new mime types. I was able to add a mime type for PDF files very easily. This tool moves adding modules and setting mime

types into the realm of the Junior SA. As long as the Junior SA knew what was needed, they would be able to add it.

Another Apache tool lets the SA set up options for individual directories on the server. This allows the SA to quickly see what options have been set and change them if needed. I used the tool to set up some restricted directories that only local users could have access to. Putting all of the directory information in one place would be of benefit to both Senior and Junior SA's. You can also view all Apache Log files from a central location and through the consistent user interface.

A final option in Apache allows you to directly edit the Apache configuration file. This is not a graphical tool, but simply a text editor that allows you to make changes directly to the file. There is both a pro and a con to this. On the plus side, you can still make command line changes to the configuration file, and they will be valid. On the con side, you still have to know how to edit the file by hand.

Webmin also allows you to set up other kinds of servers such as DNS, FTP servers, SendMail, PostFix, and Squid. In each case, basic options are configurable through menu items, and the configuration files can be edited in a text window. The advantage to using this tool is that all service information is gathered in one location, and can be accessed in a common way.

The Network Configuration Tab is used to configure NFS, as well as to set up your inet.d or xinet.d, iptables and network interfaces. Normally, the configuration files are scattered all over the file system, and can be hard to locate. I did not have NFS setup on my computer, so I used Webmin to do it. I was able to easily create exported directories and set up the sharing and security information. I used the network interface tool to verify the information about my Ethernet connection, as well as check my routing information. I used the graphical hosts tool to set up my hosts file with an alias for all locally connected machines. This was all extremely easy, and did not require me to read any documentation or do a whereis to find configuration files. Again, senior SA's who have done this for years will be more comfortable with the command line interface, but Junior SA's will prefer the ease and consistency of this tool.

The Hardware Tab is used to set up partitions and the boot loader. My guess is that this will be less used than some of the other tabs. This section does have a printer configuration tool that is extremely useful. I had had trouble getting my Linux box to see the printer that I had installed on my Windows computer. Using the Printer Configuration Tool, I was able to automatically connect to and configure the printer. This could save a great deal of time for Junior SA's.

### **Lessons Learned/Problems**

The graphical nature of Webmin is also a plus for both experienced and new SA's. By seeing all of your options at once, you will be less likely to forget to perform key tasks. It also provides nice visual summaries of the entire system. Webmin is also consistent across Unix platforms. Webmin is supported on 47 different versions of Unix and Linux, including all of the major players. If you are in a mixed Unix environment, you will most likely have to perform any given administrative task differently in each OS. If you use Webmin to do all administration, you will have a consistent interface for across all platforms. The situation in which Webmin would not

work would be if the Web server were down. If the Web server is down, you have no way to get into Webmin at all.

### **Final Results/Recommendations**

I would strongly recommend the installation and use of Webmin. It is available at no cost, and could greatly increase the efficiency especially of Junior SA's. The scope of services available to the SA through Webmin is impressive. Almost any task that can be accomplished through the console can be preformed remotely. Webmin provides a central access point for configuration and installation information, and will save time spent searching for the right configuration file.

Security is always a concern with a product like this. You are in essence allowing configurations to be changed via the public Internet. Webmin addresses these concerns with the ability to restrict access to the service to specific IP addresses. Another nice security feature is that Webmin allows you to set up users to do specific administrative tasks. Several articles in the LISA proceedings addressed the issue that you often do not want to give full root access to a Junior SA who is only in charge of adding new accounts. With Webmin, you can have many users and groups that are not root, but that still can accomplish tasks that normally require root access.

Webmin exceeds the requirements for a tool to do some handholding for Junior SA's. All SA's would benefit from its tight integration of all administration tasks into a simple, consistent graphical environment. I would highly recommend this product.

I spent 3 hours researching the product online. Downloading and installing the program went quickly and only took an hour. I spent 8 hours over several days exploring all of the features of this product. This was more than just product exploration. I actually used Webmin to configure and maintain my home system, and plan to continue to use it. Writing and editing this document took an additional 4 hours, for a total project time of 16 hours.

**Product:** zenTrack

**Product Type:** Ticket Tracking Software

**Author:** C. Judith Nyabando

---

### **Problem Background**

System administrators are often faced with the challenge of keeping up with customer requests. This becomes more difficult in a large organization. Ticket tracking software helps automate the process of managing customer requests by tracking a request from the time it was logged into the system to the time it was closed.

### **Product Placement**

zenTrack is a ticket tracking system for tracking projects, requests, and gathering information. The software is not meant for system administrators alone. It can be used by other departments in the company to track their projects.

There is a demo available on the zenTrack website that potential users can use to test the software but I wanted to install it to test the installation process as well.

### **Installation Overview**

zenTrack is web based, and it uses a database and php. The developers of zenTrack claim that zenTrack should work with MySQL, Oracle 8i, Oracle 9i, Postgress, Sybase and SQLServer. The operating systems supported are Red Hat Linux, Windows 2000, Windows NT, Slakeware Linux and Debian Linux. The supported web servers are Apache 1.3, Apache 2.0, and IIS. For this project I used MySQL, Windows 2000, and IIS.

The first thing I did was to download php, MySQL, and zenTrack binaries for Windows. I then installed and configured php and MySQL. To configure php I had to edit `php.ini` file to reflect the following:

```
Commented error_reporting = E_ALL & ~E_NOTICE
Verified that session.save_path was set to a valid directory
short_open_tags = On
```

zenTrack has two main directories: `zentrack\includes` and `zentrack\www`. I placed `zentrack\www` in the `wwwroot` directory, and `zentrack\includes` in Program Files directory and renamed it to `zen-includes`. I then set the appropriate NTFS permissions on the contents of these directories.

The next step was to create a database instance: `CREATE DATABASE zentrack;` and then load `build_mysql.sql` and `seed_mysql.sql`. I created a new user for this database:

```
GRANT ALL PRIVILEGES ON zentrack.* TO admin@localhost IDENTIFIED BY
'admin'
```

I then configured `www\header.php` to reflect at least the following:

```
$Db_Instance = zentrack
$Db_Login = admin
$Db_Pass = admin
$Db_Host = localhost
```

The last step was to configure IIS so that the contents of `www` can be displayed on the web.

### Lessons Learned/Problems

The installation process did not work. The database was running and I had created a user account (`admin, admin`) on the zenTrack database instance but the software could not connect to the database. I posted questions on the Zentrack discussion board and followed the instructions they gave me but problem did not get resolved. Below are the errors messages displayed every time I try to access the `www (zentrack)` website:

```
Notice: Undefined index: login_name in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_id in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_level in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_inits in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_bin in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_mode in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: login_messages in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: project_mode in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined index: ticket_mode in C:\Program Files\zen-includes\session_start.php on line 51
Notice: Undefined variable: lang in C:\Program Files\zen-includes\headerInc.php on line 65
```

The demo on the zenTrack website presents zenTrack to be very effective in meeting users needs. There are basically two kinds of users who can use the software: administrators and regular users. Regular users have privileges to create new tickets, e.t.c. Administrators have the privileges that the regular users have as well as privileges to create user accounts and manage the database. Figure 4.2 below shows details about tickets currently in the system.

D	Title	Pri	Type	Opened	Owner	Bin	Time
61	sadfas	Critical	Service	04/16/2003	n/a	Tech Support	2.3 hrs.
48	Bug1	Critical	Bug	04/16/2003	USER	Tech Support	19.1 hrs.
44	blah	Critical	Bug	04/15/2003	USER	Tech Support	26.8 hrs.
23	test	Critical	Bug	04/14/2003	USER	Tech Support	53.5 hrs.
35	Raah!!!	Medium	Bug	04/15/2003	n/a	Tech Support	36.2 hrs.

Figure 4.2 Tickets in the system

Figure 4.3 shows a new ticket that has been created. The ticket has high priority and it has been assigned to zenTrack administrator. The left and right columns display the options that can be applied to this ticket. The close option will close the ticket; move will move the ticket to another bin (department).

LOG	<table border="1"> <thead> <tr> <th>ID</th> <th colspan="2">TITLE</th> </tr> </thead> <tbody> <tr> <td>64</td> <td colspan="2">kl</td> </tr> <tr> <th>ELAPSED</th> <th>OPENED</th> <th>DEADLINE</th> </tr> <tr> <td>0 hours</td> <td>04/17/2003</td> <td>05/17/2003 00:00</td> </tr> <tr> <th rowspan="2">OPEN</th> <th>PRIORITY</th> <th>OWNER</th> </tr> <tr> <td>High</td> <td>zenTrack Administrator</td> </tr> </tbody> </table>			ID	TITLE		64	kl		ELAPSED	OPENED	DEADLINE	0 hours	04/17/2003	05/17/2003 00:00	OPEN	PRIORITY	OWNER	High	zenTrack Administrator	REJECT
ID	TITLE																				
64	kl																				
ELAPSED	OPENED	DEADLINE																			
0 hours	04/17/2003	05/17/2003 00:00																			
OPEN	PRIORITY	OWNER																			
	High	zenTrack Administrator																			
RELATE				PRINT																	
MOVE				EMAIL																	
CLOSE				EDIT																	

Figure 4.3 new ticket

### Final Results/Recommendations

I discovered from the Zentrack discussion board that a lot of people who were trying to install the same software encountered the same problems I encountered. It appears as if some people got it working and some are still struggling with the same login problem. I also think that lack of experience with php makes it even more difficult to debug the problems.

### References

- zenTrak. 30 April 2003 <<http://zentrack.phpzen.net/>>
- PHP Group. 30 April 2003 <<http://www.php.net/downloads.php>>
- MySQL. 30 April 2003 <<http://www.mysql.com/>>

# *E-Mail*

**Product:** IMail 7.15  
**Product Type:** E-mail server  
**Author:** Trey Buck

---

### **Problem Background**

Mail services are an integral part of any computing infrastructure. In many cases, it is the glue that holds organizations together. Often given the highest priority of all the critical services, it is important that it have high availability.

However, mail servers are often complicated entities and are intimidating to inexperienced system administrators. Sendmail is lightweight but is notoriously complex to configure. Exchange is simpler to configure, but consumes large amounts of resources. Mail servers that combine simple configuration with a lightweight process are somewhat rare.

### **Product Placement**

IMail is a commercial mail server that runs on MS Windows platforms. It provides services for SMTP, POP, IMAP, WHOIS, finger, web messaging, web calendaring, and monitoring. Other features include support for virtual domains, mailing lists, virus scanning, server-side rules, SSL, load sharing, and LDAP. IMail's list of features makes it attractive to large organizations.

IMail is designed to be a robust mail service that has low administration costs. It has a reputation of being simple to install, configure, and maintain. It also scales well, housing up to 100,000 users per server and able to process one million messages per day. The low administration cost and scalability makes it attractive to organizations with a limited IT staff.

### **Installation Overview**

The installation process is very simple and follows a typical installation format. The steps to install and configure are listed below.

- Run the executable file `imtm_x86.exe` (We used an evaluation copy; the full version will have a different filename).
- The install program will first prompt for the FQDN of the mail host. In our case this was `rissserver.ris.com`.
- The installer asks what user database IMail should look for. In this example, the Windows NT user database was selected. The other two options are the IMail database or another ODBC compliant database.
- The installer asks what directory to install the server to. The default is `C:\IMail` and this is what we chose.
- The installer will next ask what Program Folder to list the IMail server under. We chose the default of `Programs\IMail`.
- Choose whether or not to install default SSL keys for the web services.
- Choose how email relaying should be handled. At install time, only two options are given: `relay all main` or `relay no mail`. More options are available after installation.
- Choose services to start by default. We chose to start all services by default.
- Finish

If the appropriate services were set to start by default and IMail is using a Windows NT user database, the mail server is functional. There are however, a few more tasks that need to be completed. These include configuring how services will be monitored and access control.

Service monitoring is very flexible but is not enabled by default. Each service is logged individually and the logs can be directed to a text file, the Windows application log, or a dedicated log server. Monitoring is available for every service IMail provides and also for other non-IMail services including: WWW, FTP, Telnet, DNS, NNTP, default gateway, and disk usage. IMail also supports notification via email or pager.

Access control should be configured as well. Access control can be set for the monitoring service, POP3, and SMTP. For each of these services, the default access rule can be set to either grant or deny all hosts. Exceptions to the rule are granted to individual hosts or subnets. In addition, SMTP relay can be set to no relay, local users only, local hosts only, all hosts, or individual hosts and subnets can be specified.

The following table lists what services were tested.

<u>Service</u>	<u>Purpose</u>	<u>Tested</u>	<u>Working</u>	<u>Required Configuration</u>
Finger	Return user information	Yes	Yes	No
IMAP4	MUA	Yes	Yes	No
LDAP		No	--	--
Monitor	Service monitoring	Yes	Yes	Yes
Password	Change password daemon	Yes	Yes	No*
POP3	MUA	Yes	Yes	No
SMTP	MTA	Yes	Yes	No
SysLog	Error/Status logging	Yes	Yes	No
Web Calendaring	Calendaring	Yes	Yes	No
Web Messaging	MUA	Yes	Yes	No
Whois		No	--	--

\* Cannot change passwords if using WindowsNT user database

### **Lessons Learned/Problems**

No problems were encountered at all during installation or configuration. The only issues discovered were related to the web interface. The first concerns the ability for a user to change their password. If the IMail server is authenticating from a Windows NT database, the user will be unable to change their password via the web interface. The option to change the password is still presented however. This can be fixed by simply editing the appropriate HTML file. The second issue is that the web mail and web calendaring are not tightly integrated when using the default web site template. It must be noted that there are alternate web site templates that are available both freely and commercially that may address these issues.

**Final Results/Recommendations**

Installing the software is simple and can be done in less than ten minutes. Configuration not a difficult task, thanks in part to the IMail Administrator applet. The Administrator's interface is very simple and intuitive to anyone familiar with Microsoft Management Console snap-ins. It's ease of use means less time reading documentation and more time actually administrating the service.

The IMail server is recommended to any organization, particularly those that run a Windows shop. It is also based upon open standards, making it suitable for use in UNIX & Linux environments as well.

**References**

Ipswitch, Inc. 2003. 30 April 2003. <<http://www.ipswitch.com>>

**Product:** Libeum Help Desk

**Product Type:** Web based help desk

**Author:** Robert Nielsen

---

### **Problem Background**

To provide a high level of support for users, it is essential that good communications channels be made available. Depending on the size and type of organization, the primary communication channels may vary greatly. In situations that involve small, low structure environments, the phone and voice mail may be enough to handle most or all of the communications requirements. With larger and more diverse user populations, support personnel often find the need for a more structured system for managing communications. Many times the next step from phones will be E-Mail communications. E-Mail is often better than voice mail because it is less time restricted and allows for greater reflection on the content. It also provides a record of communications that simply is not available when using the phone. This creates a better situation for the person making the request, as requests are less likely to be forgotten or misplaced, and a better situation for the support staff as it allows them to more easily document requests received and set priorities.

Though an E-Mail based communications system has many strengths, it also has some weaknesses. One of the more common problem issues in many places is that E-Mail isn't properly functioning. If E-Mail is the primary communications channel for users needing assistance, in the event of an E-Mail system problem they won't be able to report the problem. In cases where there are many users submitting requests and the need for communications with other parties on subjects unrelated to user requests, managing the volume of mail can also become an issue. A shared E-Mail address that is dedicated to support request can help in this situation, but setting up such an address isn't always a viable option. E-Mail also doesn't allow easy for generation of reports and searching through old E-Mail messages to find a previous request or response can be time consuming and frustrating.

To address these concerns, more and more support providers are adding new technologies to help manage user requests. One of these additions is the use of help desk software. Some help desk packages are essentially front ends to a database that still require someone to answer a phone or read an E-Mail so that they can enter the request. Others have added a front end so that the person making the request can enter their request directly, reducing a portion of the overhead associated with generating a helpdesk request. One of the methods of allowing users to directly enter request is by use of a web page. This generally removes the need for any special software on the client system as most modern systems include a browser.

In reviewing the options available, there are several things that one might consider. The most obvious consideration is often cost. Another consideration might be the ability to secure the software so that only authorized users can enter requests. To help with the problem of locating old requests and responses, a good search system might be of concern. Size, speed and ease of use might also be taken into account when reviewing the help desk software options available.

In an attempt to locate a software package for the ETSU Computer Science Department, each of the above was considered. Several software packages were located that met some of the

requirements, but only a few met the primary requirement for this site. The primary attribute the Department required was that the software was free. Of the located packages that were free, one stood out as appearing to meet all of the other requirements as well. The package Liberum was selected for testing and possible deployment in the Computer Science Department.

### **Product Placement**

Not only was Liberum free, but it was also small, fast, easy to configure, web based, and was designed to run on the primary web platform in use by the Department. Liberum was noted as being available for use under GPL licensing. The download for the asp and html pages that comprise the web based package was under 1 Mb. Liberum was designed as a web interface to the help desk that was compatible with every tested browser. For security, Liberum offered the Department a number of options for setting limitations on who could enter requests. It allowed selection of database authentication or selection of NT authentication. It also would allow the Department to select between using an Access database or a SQL Server database for storage. Additional features that Liberum was found to offer include:

- the ability to categorize requests
- the ability to designate a particular “support rep” as handling one or more request categories
- the ability to generate E-Mail notifications that would be sent to the designated support rep
- the ability to generate E-Mail confirmations that would be sent to the person making the request
- the ability to generate E-Mail notices upon closing a request
- allowing users to review and update their support requests
- menu driven navigation for ease of use
- the ability to build a searchable “Knowledge Base” of problems and resolutions
- the ability to generate reports on requests
- the ability to configure support for pagers
- Mail handling via CDONTS, Jmail, ASPEmail, or ASPmail

### **Installation Overview**

Initial setup of Liberum started with unzipping the downloaded package. Next a folder called “helpdesk” was created under the root web on CSCIDBW. The contents of the package’s “www” folder were then copied into the “helpdesk” folder. A second folder was created on CSCIDBW to hold the database and the contents of the “db” folder were copied into place there. It was then necessary to manually edit the file settings.asp to designate the database type and location. Once this had been edited, the remainder of the configuration took place via a web browser. The browser was pointed at <http://cscidbw.etsu.edu/helpdesk/setup.asp>. This configuration page was password protected and required that the user logon before allowing access. Once logged in, the software presented a menu of options related to configuration. The most important of these was the “Configure Site” option. By using this option, the administrator could do site configuration such as naming the site, setting the primary administrator name, setting the E-Mail handling method, selecting the authentication method, etc. Other options from the configuration menu allowed for testing the configuration, configuring the format of outgoing E-Mail messages, managing users, managing request categories, managing priorities, managing request status options, generating reports, etc.

**Lessons Learned/Problems**

With the configuration completed, it seemed best to get the opinions of an actual user on how easy the system was to navigate. Melody Henry, one of the Computer Science Department secretaries, went to the site and completed the user registration. Once registered, she created a request for a software install and submitted. She then reviewed the request to verify it was as she desired. Her overall impression of the system was good. She did recommend that users be sent a memo with a quick overview of the system so it would be easier to “get started” as a user. She seemed impressed by the automatic E-Mail system that could keep her “up-to-date” on what was happening with her request. With her request made, one of the support reps received an E-Mail notice. He checked the web site and found a new request that had been assigned to him. He then proceeded to fulfill the request and returned to the helpdesk software and closed Melody’s request. Once this was done, Melody received an E-Mail notifying her that the work was complete.

**Final Results/Recommendations**

Based on the initial testing, Liberum seems to offer all of the features currently needed by the Computer Science Department for a help desk implementation. The only items of concern found to date are that a user could, intentionally or accidentally, enter an incorrect E-Mail address and that the Knowledge Base might present a privacy issue. The only bug noted in the software to date is the requirement of an E-Mail server address, even when the server is unneeded for functionality. Though not a problem in our setting, it also appears that domain authentication for users must either use the local system domain, or the domain which has been selected for web server and ftp server authentication. Attempts to designate a third domain for authentication failed. The current plan is to continue testing Liberum with a limited number of users, so a greater degree of confidence in its abilities can be built. If it continues performing well through testing, it is expected to be made available as the primary communication channel for problem submission and resolution in the Computer Science Department.

**References**

Luxem, Doug. 29 April 2003 <<http://www.liberum.org>>

**Product:** Microsoft Exchange

**Product Type:** E-mail server

**Authors:** Adam Berry, Kao-Yee Chua, Sai Divvala, Robert Nielsen, Judith Nyabando

---

### **Problem Background**

E-Mail is generally considered one of the two most important services offered to users. Because of this, it is valuable to have experience with the process of building a mail server, such as Exchange, and configuring it to interact with other mail servers. When corporate information is handled by a mail server, it can also be important to implement security so as to provide some degree of confidentiality for the information being processed. Security measures such as SSL should only be a part of the corporate policies for protecting E-Mail, because SSL is only of use when dealing with mail that never leaves the corporate Intranet. For transactions that traverse the Internet, mail should be secured by use of an encryption tool like PGP.

### **Product Placement**

To gain experience with the installation and configuration of mail transport agents and mail user agents, our group was assigned to install and configure three Microsoft products: Exchange 2000 was the mail transport agent, and both Outlook 2000 and Outlook Web Access (OWA) were the mail user agents. We also were assigned to configure OWA and to use SSL to provide for secure communications between the client and the server. Lastly, we were to integrate our e-mail setup with the other two groups so that the systems could simulate a real world e-mail environment.

### **Installation Overview**

The first portion of the project involved installing the software. We began by installing a clean copy of Windows 2000 Advanced Server on our machine. All of the settings for the Advanced Server installation were the defaults. Once the operating system installation was complete, the server (ms.example.org) was also configured as a domain controller. The next step we took was to install Windows 2000 Service Pack 1 on our server. With the operating system setup complete, we next configured the DNS server on our system. Next, Exchange 2000 and Exchange 2000 Service Pack 1 were both installed on our server. At this point, a new Exchange organization was created for our server, with the organization being called “mailroom”. This setting was the last portion of the install that required any user intervention. With the installation complete, we created users in our domain. When creating the users, the wizard prompted us as to whether or not to create an Exchange mailbox and mail alias for the user that was being added. We wanted to use a namespace that was simple and clear about who the owner was. Knowing that each person in our class had a unique last name, we elected to use last names for our usernames. While using last names might not be a good choice in another setting, it seemed reasonable for the purposes of this project.

Exchange Server comes fully functional out of the box. Once installed, we were able to access and use e-mail services with both Outlook Express and OWA. However in order to allow secure communications we had to perform several additional steps. The first of these steps was to install Certificate Services so that Windows could generate and maintain certificates for secure e-mail communications. Because this machine was the domain controller on our domain example.org, we chose the option for “Enterprise root CA”. This option designated the system to

be the most trusted certificate authority in the hierarchy of certificate servers that could exist on a domain. Our second step was to configure OWA to use SSL for secure communications. In order to do this we had to request a new certificate from our certificate authority. The certificate was requested using the default options, as we were our own certificate authority. Anyone needing a real certificate would have sought one of the well-established certificate authorities on the Internet. After the real certificate authority filled the request, the administrator would then import the certificates into their server for authentication. Similarly, our third step was to configure OWA for SSL by using the Internet Information Services MMC to initiate the IIS Certificate Wizard to select our new certificate and finalize its' implementation. The final step in configuring SSL was to use the Internet Services Manager to set our web access to require a secure channel. At this point we successfully tested OWA over SSL using Internet Explorer. With this certificate in place, IMAP4, POP3 and other mail exchange protocols could be configured to use the same certificate for secure communications using the Exchange System Manager.

The last step of our project build was to install Outlook 2000 and to integrate it with Exchange 2000 using MAPI. The Outlook 2000 installation was done by using the Office 2000 CDs. To use Outlook 2000 and MAPI, you should select the "Corporate or Workgroup" option during the Outlook installation. Since we had not made this selection during the install, we had to use the "Reconfigure Mail Support" option which we found under the Options section of the Tools menu. After reconfiguring the mail support for MAPI, we were able to successfully test our Outlook client.

### **Lessons Learned/Problems**

With our project system set up and working, we installed and configured a similar system with a different domain to simulate integration with other e-mail servers. In doing so, we encountered some issues. The first issue dealt with messages not being properly sent because Exchange improperly handled certain DNS records. The second issue was evidently caused by a server crash that occurred when messages were in the queue. This crash led to the Exchange services not properly restarting. These issues were resolved by applying service pack three for Windows 2000 and service pack two for Exchange 2000. Thus we recommend installing the latest service packs for both Windows and Exchange before attempting to initiate inter-server mail communication.

### **Final Results/Recommendations**

In the end, our server was able to successfully communicate with another server as well as with all clients tested. We were also able to add another server to our network and perform mail transfers to and from that system. From our experiences we would recommend Exchange as an E-Mail server platform. It was easy to setup, easy to use, and it provided all desired functionality.

**References**

- Exchange 2000 Server Support Center. Microsoft Corporation. 2003. 29 April 2003.  
<<http://support.microsoft.com/default.aspx?scid=fh;EN-US;exch2k>>
- Fugatt, Mark. "Securing Outlook Web Access using SSL". MExchange.org. 6 August 2002.  
29 April 2003 <<http://www.msexchange.org/tutorials/MF004.html>>
- MExchange.org. 2003. 29 April 2003 <<http://www.msexchange.org/>>
- Support - Microsoft Exchange Server. Microsoft Corporation. 2003. 29 April 2003.  
<<http://www.microsoft.com/exchange/support/default.asp>>

**Project:** Postfix Mail Server Project

**Authors:** Mario Hankerson, Amanda Hickman, Todd Franklin, Steve Fritts

---

### **Problem Background**

Our assignment was to install a mail server using Postfix, the IMAP protocol, and Pine and communicate with other groups in the class.

### **Project Goals**

Our primary goal was to see if we could get these three elements to work together to be able to send and receive mail from the other two groups. RedHat's website claims that Postfix is "very fast" and "is extremely light on its usage" of server hardware. The IMAP protocol allows mail folders to be stored on a central server (instead of on a user's local machine). IMAP allows us to:

- access mail folders from any machine, anywhere (this requires the IMAP client to be installed)
- manage multiple mail folders belonging to multiple e-mail accounts from a single client
- switch mail clients (Netscape, pine, Eudora) at will, and automatically carry all of our mail folders with us

Finally, we used Pine as our IMAP client which allowed us to compose, send and receive e-mail. Pine 4.x versions offer support for online IMAP folder access.

### **Project Details**

We started the installation with a clean install of RedHat 8.0. We simply included the Postfix and `imapd` packages as part of our RedHat 8.0 operating system installation. Pine was not included in RedHat8.0 (although it has been included in previous RedHat versions), so we downloaded the Pine RPM off the web and installed it separately.

Configuring Postfix, the IMAP server and Pine were not quite so simple. We encountered several problems trying to get these products to work. We will describe the Postfix configuration first.

Postfix installs four configuration files in the `/etc/postfix` directory:

```
install.cf
main.cf
master.cf
postfix-script
```

`install.cf` contains initial settings for Postfix-these were included with the installation. `main.cf` is Postfix's primary configuration file. `master.cf` is Postfix's master process configuration file. Each line in this file describes how a mailer component program should be run. The `postfix-script` file allows Postfix to be used with the Linux operating system. All configuration changes necessary to get our Postfix installation working were made to the `main.cf` file (we were able to leave the `install.cf`, `master.cf` and `postfix-script` files as they were). Below are our sample configurations for `main.cf`:

```

queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
mail_owner = postfix
myhostname = os2.example2.org
mydomain = example2.org
myorigin = $myhostname
inet_interfaces = all
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
disable_dns_lookup = yes

```

IMAP also had files that had to be configured. Changes had to be made to the `/etc/inetd.conf` file.

Further changes had to be made to the default Pine configuration. Pine configuration does not require editing of files--it can be done through Pine's user interface. This is accomplished by pressing the 'S' key (Setup), then 'C' key (configuration). We had to configure our mail server name, user names, and mail folder directories in order to get Pine to work.

In addition to these changes, we had to modify other files that are part of Linux. These included:

```

/etc/hosts
/etc/resolv.conf

```

#### **/etc/hosts**

```

127.0.0.1          localhost.localdomain  localhost
10.11.11.20       os2.example2.org      os2
10.11.11.111     os1.example1.org      os1

```

#### **etc/resolv.conf**

```

nameserver        10.11.11.3

```

After all configurations were complete, we were able to start the Postfix server (the command is `postfix start`). Once Postfix was running, we were able to run Pine and attempt to send and receive e-mail.

#### **Lessons Learned/Problems**

After completing this project, we were able to install Postfix, IMAP and Pine and configure them to work together so that we could send and receive mail from users on other mail servers. In doing reading prior to the actual labwork, we found that each of these three products can be used with other mail products that are available for free. For example, Netscape or Mutt can be used to replace Pine, and POP3 can replace IMAP, and Sendmail can be used in place of Postfix.

We encountered several problems while working on the project. In order to use Postfix, Sendmail must not be installed. It must be removed prior to the Postfix installation or it will botch the installation. We had numerous problems with the Postfix setup because Postfix was expecting certain files in certain locations and they weren't there. Also, Pine was dropped from the available packages included on the RedHat 8.0 CDs. This software must be downloaded and

compiled in order to be used, (precompiled versions are available). Users unfamiliar with compiling software in Linux may want to try the precompiled versions.

### **Final Results/Recommendations**

After configuring the files listed above, we were able to send and receive mail to and from ourselves (via user accounts we set up in Pine). Sending and receiving mail to accounts in the other groups required more changes, specifically to Linux's networking files.

As mentioned above, Postfix is supposed to be a faster alternative to sendmail. Our project did not test this. This would be an interesting project. Another interesting project would be to do comparison with some of the other mail products available for Linux (POP3 versus IMAP, Mudd versus Pine, Postfix versus Sendmail) and find the pros and cons of each setup.

### **References**

- "RedHat Postfix HOWTO". 2000. 9 Apr. 2003 <<http://www.redhat.com/support/resources/howto/RH-postfix-HOWTO/book1.html>>
- "RedHat Postfix HOWTO". 2000. 9 Apr. 2003 <<http://www.redhat.com/support/resources/howto/RH-postfix-HOWTO/c108.html>>
- Jao, David. "IMAP on Linux: A Practical Guide". Linux Gazette. 1998. 9 Apr. 2003 <<http://www.linuxgazette.com/issue35/jao.html>>

**Project:** Sendmail, Cyrus Mail Server Project

**Authors:** Trey Buck, Lingling Duan, David Frazier, Rick Simons, Gunter Wambaugh

---

### **Problem Background**

Our assignment was to install a mail server using Sendmail and Cyrus IMAP. We then had to use the mail server to communicate with other groups in the class.

### **Project Goals**

Our goal was to use Sendmail and Cyrus IMAP to create a mail server that would allow us to create user accounts and send mail to two other groups in the class.

### **Project Details**

Our first task was setting up the operating system on our lab computer. We choose to install Red Hat 8.0 and performed a custom setup where we installed SendMail, Cyrus SASLdb, and Ximian Evolution. Our next task was to download Cyrus IMAP. Our first attempt involved downloading the source tarball from the Cyrus project homepage. This installation was complex, but with no apparent problems. We located several documents on how to configure Cyrus IMAP. The different sets of instructions, however, often had conflicting information in them. The two top contenders were the documentation that came with Cyrus, and third-party documentation found at <http://en.tldp.org/HOWTO/Cyrus-IMAP.html>. We started off with the internal documentation, switched to the external when we had unresolved questions, and switched back to the internal to finish up. This consisted of editing several configuration files within Cyrus. The Cyrus master process was then started, and it was verified that we could telnet to both the IMAP and POP ports of the machine. Our next step was to use cyradm to create mailboxes for each team member. We then customized Evolution to point to the new mail server. When we tried to use Evolution, we could not get authenticated to our mailboxes. Several attempts to fix this authentication were unsuccessful. An internet search on the error messages we were getting returned that many had the same problem, but no one had any answers.

At this point we decided to see if there were any RPM's available. We found one from what seemed to be a reliable source, <http://home.teleport.ch/simix/>. Several Web pages that we found pointed to this site, and we performed a clean OS install, then downloaded and installed the RPM with no problems. We started the server, as before, and again tested to see if we could telnet to the IMAP and POP ports, and found that we could. We were able to use imtest to authenticate to the sasldb. Things were looking up until we tried to create the mailboxes. We found that cyradm would not run, complaining that it could not locate the perl modules that we had just installed. Several of us tried to use symbolic links to get cyradm to see the modules while others searched the Web for a solution. Again, many seemed to have this problem, but few had any answers. We finally found an answer that pointed to a bad RPM, which was available online <http://www.cencula.com/cyrus.html>. We uninstalled the RPMs and started over yet again.

We attempted to install the IMAP product from source again, trying a new set of documentation that we had found. This time, the install seemed to work. We could telnet to the port, authenticate, create mail boxes and when we set up Evolution again, we could actually get into

our mail boxes. We then configured sendmail as our Mail Transport Agent. Finally, we configured our hosts file to be able to see the other groups. We were certain the mail system would work and it did, up to a point. Exchanging mail within the local system works beautifully. The catch, of course, is that we cannot exchange mail with other servers.

### Lessons Learned/Problems

Our problem seems to lie somewhere within the sendmail configuration. For unknown reasons, and despite determined efforts, we were unable to coax sendmail into communicating with the other two servers. A breakdown of working and nonworking components follows:

Cyrus	Working
Ximian	Working
Sasl	Working
BerkelyDB	Working
Perl 5.8 & Perl 5.12	Working
Web Client	Installed but not configured
Able to sent mail on same machine.	Working
Able to sent to other machines	Not working. Probably a sendmail issue.

The biggest problem our group faced was lack of a clear leader. Except for Rick and David, the team members had never worked together before. Many of the team members did not know each other before this project, and the lack of a leader meant that there was no clear work agenda. If someone had time, they came in and worked on the configuration as much as possible. There was some documentation kept, but not all team members knew about it, and it was not always clear what was going on. Therefore, two different groups using two different sets of documentation completed our first install. Another problem made worse by lack of communication, was that some team members were Linux experts, while some were much more comfortable in Windows. Instructions written in Windows did not always translate seamlessly into Linux.

Another problem plaguing this group was that everyone was on a vastly different schedule. Most team members are employed full-time, and some live far from campus. There was not a single meeting time that would have allowed all members to be present. This only made our communication problems worse. Another major factor in our failure was a continuing drop in enthusiasm for the project. This was made worse by seeing the other teams breeze through their installs in one setting, mostly performed by one individual. For a one-point project, the time we were putting in was just not worth it, and the more time we put in, the worse morale became.

Things we could have (and should have) done differently. First and foremost, a team leader needed to be appointed. Someone needed to be aware of what was going on with everyone. Second, we needed to take better notes. We needed to have a log that everyone knew about, and to make relevant, clear entries whenever we tried something. This task was also affected by watching the other groups. We saw how easy their projects went, and assumed that ours would go the same. For a simple install, documentation is not as important. We realized too late that any install can be problematic, so we needed to be tracking what we had tried, so we would not try the same thing again.

On a positive note, we knew when to cut our losses and start over again. We learned about mail systems and how they operate, group dynamics, and how to resolve conflict when it crops up. We did get Cyrus up and running, which we considered to be the major task of our project. When this was accomplished, we felt that we had more than earned our project point especially considering the difficulties we ran into. Our installation was more like real life, where conflicting information and guides are prevalent and we hope the other groups can learn that lesson from our failure.

# *Case Studies and Other Projects*

**Project:** Building Cables for Wilson-Wallis CSCI2150 Area

**Author:** C. Judith Nyabando

---

### **Problem Background**

The CSCI2150 area in the Wilson Wallis lab had no cables to connect the computers to a patch panel. Mr. Steven Jenkins and Mr. David Tarnoff wanted the cables build to meet the networking lab requirements for the CSCI2150 Computer Organization course.

### **Project Goals**

The goals of this project were to build twisted pair cables for the CSCI2150 area in the Wilson Wallis lab and use them to connect the computers to the patch panel. The subsequent sections discuss the steps I took to build the cables.

### **Project Details**

Required tools:

- Twisted pair cable
- RJ-45 plugs
- Cutter/stripper
- Crimper
- Cable Tester

### **Making the cables**

This section discusses the steps I took to make the cables. The process of making cables is simple but it requires a lot of concentration to make sure that the cables are made right on the first build, thereby saving resources. The steps of making a twisted pair cable are discussed below:

Measure and cut the cable to the desired length. Strip one end of the cable using a cutter or stripper. Untwist the twisted wires and arrange them according to the appropriate standard for connecting systems to a hub. Below is the wiring standard:

Pin 1 (Data Send +)	_____	Orange/White
Pin 2 (Data Send -)	_____	Orange
Pin 3 (Data Receive +)	_____	Green/White
Pin 4 (Reserved)	_____	Blue
Pin 5 (Reserved)	_____	Blue/White
Pin 6 (Data Receive -)	_____	Green
Pin 7 (Reserved)	_____	Brown/White
Pin 8 (Reserved)	_____	Brown

Trim the ends of the wires to make them even and push the wires into a RJ-45 plug. The wires should layout evenly against the end of the plug. If the wires are even, they can slide smoothly into the RJ-45 plug. If not it will be difficult to get the wires laid out evenly against the end of the RJ-45 plug. The above steps should be repeated for the other end of the cable. The arrangement of the wires should be exactly the same in the RJ-45 plugs on both ends of the cable. If the arrangement of the wires on one end is in the opposite direction of the arrangement

on the other end, then a crossover cable has been made. Verify that the wires are in the same order on both side of the cable. If they are, crimp the RJ-45 plugs. It is important to verify that the wires are arranged correctly before crimping the RJ-45 plug because once the RJ-45 is crimped it cannot be reused. The last step is to test the cable using the Cable tester to verify that the cable was made right.

### Running the cables

The diagram below shows the layout of part of the CSCI2150 area and it shows how the cables run from the computers to the patch panel. The cabling was done with approval of Mr. David Tarnoff.

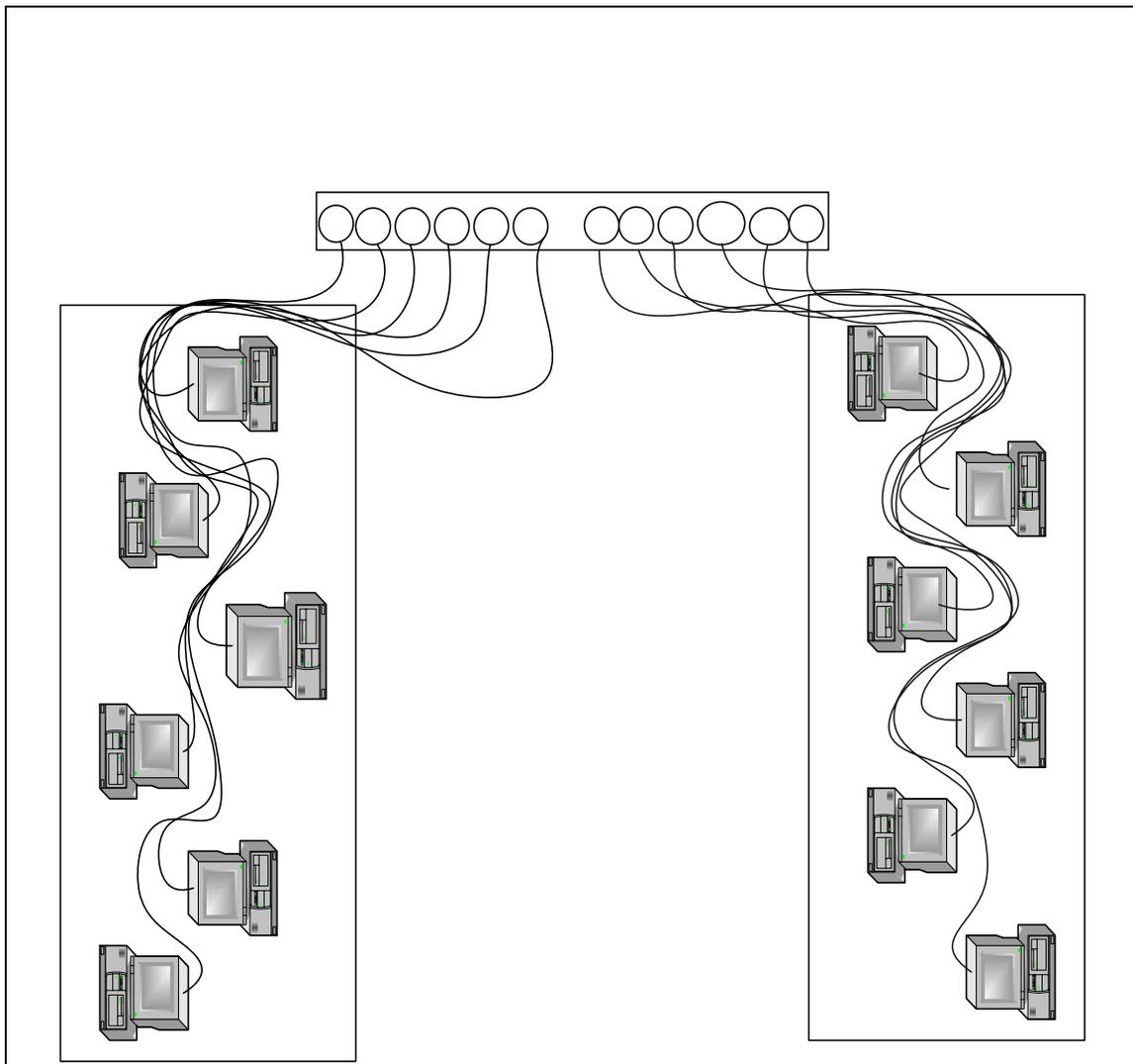


Figure 5.1: Wilson Wallace lab area for CSCI 2150

It took about nine hours to build cables for all the computers in the area. This does not include research time and preparation time.

**Lessons Learned/Recommendations**

The cables I built the first time did not please one of the customers because some of them were too short. Therefore I had to build longer cables for some of the computers. The lesson learned here is that I should have verified with all customers involved how they wanted the cables to run.

Below are other lessons learned from this project:

- Make the wires as even as possible before inserting them into the RJ-45 plug. This will make the process smooth and accurate.
- Verify over and over again that the wires are arranged in the right order and make sure your focus is not distracted as much as possible. I had to rebuild about 8 out of the 29 cables I made the first time.
- Test every cable before using it.

**References**

- "Cisco Metro 1500 Series Management Guide, Cables and Cabling". Cisco Systems. 30 April 2003 <[http://www.cisco.com/en/US/products/hw/optical/ps1923/products\\_configuration\\_guide\\_chapter09186a00800ec8a7.html](http://www.cisco.com/en/US/products/hw/optical/ps1923/products_configuration_guide_chapter09186a00800ec8a7.html)>
- Ola Eriksson. "Ola Eriksson's guide to building a twisted pair cable". 2 December 2001. 30 April 2003 <<http://mreriksson.net/miscdocs/tpcable>>

**Project:** LDAP Namespace Design for E.T.S.U.

**Author:** Rick Simons and Steve Fritts

---

### **Problem Background**

The purpose of this project is to develop a directory service schema for East Tennessee State University (ETSU) using the Lightweight Directory Access Protocol (LDAP). The LDAP standard defines network protocols used for accessing directory information as well as namespaces that define how information is referenced and organized. LDAP also sets guidelines about operations that may be performed on data in the directory. ETSU recently decommissioned their LDAP server which was running as a service on the Netscape Calendar server for faculty usage. Currently, ETSU uses Microsoft's Active Directory for its directory service.

This project was interesting because of the nature of the information and service LDAP provides. Based on personal experience, it is quite unpleasant to know who to escalate an issue to, or what group you need to talk with, but be unable to contact them due to a poor contact information infrastructure. If implemented logically, and correctly, an online directory structure can help resolve this type of problem. LDAP is one of the underlying protocols of this type of solution, so we decided to learn more about it.

### **Project Goals**

The goal of this project is to create a comprehensive, workable namespace for ETSU. This includes users (staff, faculty, students and alumni), organizations, groups, machines, and services. This includes a whitepage directory service which will support other LDAP-based address books, mail clients, etc. This type of service also helps with recurring maintenance costs by allowing the people who use the service the ability to update it. Once this is project design is complete, we will compare our final design with the existing Active Directory structure currently implemented at ETSU. Our hope is that our design will be useful, not only at ETSU but other regional universities. We intend our design to be scalable so that it can be implemented at smaller, or much larger institutions.

### **Project Details**

The information model and namespace provided by LDAP depend on entries in a Directory Information Tree (DIT). Entries are used to store attributes about an object (persons, organizations, groups, and services could be examples of objects). Each attribute consists of two parts: a type and one or more values. One or more of these attributes from an entry form the entry's relative distinguished name (RDN), which must be unique among all entry siblings. RDNS are combined with subordinates in the root of the DIT to form distinguished names (DNs), which are unique within the DIT.

The LDAP RFCs define several rules that must be present in an LDAP schema. These are described below:

- Each entry must have an ObjectClass attribute.
- Servers cannot allow clients to add attributes to an entry except under special circumstances
- An LDAP server must provide the following information about itself:

- namingContexts: naming contexts held in the server. Naming contexts are defined in section 17 of X.501.
- subschemaSubentry: subschema entries (or subentries) known by this server.
- altServer: alternative servers in case this one is later unavailable.
- supportedExtension: list of supported extended operations.
- supportedControl: list of supported controls.
- supportedSASLMechanisms: list of supported SASL security features.
- supportedLDAPVersion: LDAP versions implemented by the server.

The LDAP protocol is described using Abstract Syntax Notation 1 (ASN.1) Basic Encoding Rules<sup>7</sup>. Below are examples of entity definitions based on the ASN.1 syntax:

```
default-country PrintableString ::= "France"

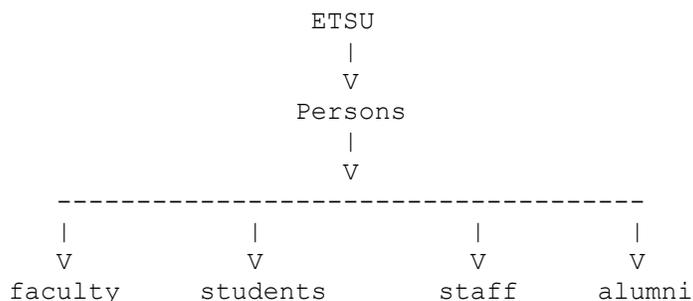
Payment-method ::= CHOICE {
    check      NumericString (SIZE (15)),
    credit-card Credit-card,
    cash       NULL }

Credit-card ::= SEQUENCE {
    type       Type-carte,
    number     NumericString (SIZE (20)),
    expiry-date NumericString (SIZE (6)) -- MMYYYY -- }
```

Entries for our namespace must use syntax similar to those described above so that any implementation of our design will be globally compatible.

### Our Directory Schema for ETSU

We decided to start with a simple design (a "whitepages" directory) and expand on it. With this design we focused on persons associated with ETSU: faculty, students, staff, and alumni.



<sup>7</sup> The complete ASN.1 definition is provided in Appendix A of RFC 2251.

Next we had to look at each group individually and see what types of information about members of these groups would be desirable in a whitepages directory. After reviewing multiple templates found online, we propose the following schema.

### ETSU-Defined Object Classes

employeeNumber – The ETSU employee/student ID. This is/can be the social security number.

OID:  
 Syntax: String  
 Source: OID  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: Order

uid - Login name.

OID:  
 Syntax: String  
 Source: UID  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: Order

userPassword – The password the user uses to authenticate

OID:  
 Syntax: String  
 Source: randomly generated  
 Min/Max: 1/1  
 Existence: Required  
 Class: Critical  
 Indexes: None

givenName - Person's first name.

OID:  
 Syntax: String  
 Source: FIRST\_NAME  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: Order

namePrefix - Person's prefix name ("Mr.", "Dr.", etc).

OID:  
 Syntax: String  
 Source : PRFX\_NAME  
 Min/Max: 0/1  
 Existence: Optional  
 Class: Normal  
 Indexes: None

nameSuffix - Person's suffix name ("Jr.", "III", etc).

OID:  
 Syntax: String  
 Source : SUFX\_NAME  
 Min/Max: 0/1  
 Existence: Optional  
 Class: Normal  
 Indexes: None

surName - Surname (person's last name).

OID:  
 Syntax: String  
 Source : LAST\_NAME  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: Order

email - Preferred email address.

OID:  
 Syntax: String  
 Source : EMAIL\_ADDRESS  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: Order

departmentNumber - HR department code. Faculty and staff only, does not apply to students or alumni

OID:  
 Syntax: String  
 Source : DEPT\_CD  
 Min/Max : 1/N  
 Existence: Optional  
 Class: Normal  
 Indexes: Order

department - HR department name. Faculty and staff only, does not apply to students or alumni.

OID:  
 Syntax: String  
 Source : HR\_DEPT  
 Min/Max : 1/N  
 Existence: Optional  
 Class: Normal  
 Indexes: None

telephoneNumber - University (office) telephone number.

OID:  
 Syntax: Telephone Number  
 Source : OFFICE\_PHONE  
 Min/Max: 0/1  
 Existence: Optional  
 Class: Critical  
 Indexes: None

postalAddress - University (office) address. For students, this is their local address, which might not be on campus. For alumni, is the home or work address, which might not be on campus.

OID:  
 Syntax: String  
 Source : BUILDING\_ROOM+" "+BUILDING  
 Source : LOCAL\_STREET\_1  
 ["\$"+LOCAL\_STREET\_2]  
 ["\$"+LOCAL\_STREET\_3]  
 ["\$"+LOCAL\_CITY  
 ["\$"+LOCAL\_STATE]  
 ["\$"+LOCAL\_ZIP]  
 ["\$"+LOCAL\_COUNTRY]  
 Min/Max: 0/1  
 Existence: Optional  
 Class: Critical  
 Indexes: Order

campusRoom - Campus building room number.

OID:  
 Syntax: String  
 Source : BUILDING\_ROOM  
 Min/Max: 0/N  
 Existence: Optional  
 Class: Normal  
 Indexes: None

campusBuilding - Campus building name.

OID:  
 Syntax: String  
 Source : BUILDING  
 Min/Max: 0/N  
 Existence: Optional  
 Class: Normal  
 Indexes: None

campusZipcode - Campus zipcode.

OID:  
 Syntax: String  
 Source : CAMPUS\_ZIP\_CODE  
 Min/Max: 0/N  
 Existence: Optional  
 Class: Normal  
 Indexes: Equality

homePhone - The telephone number of the person's residence.

OID:  
 Syntax: Telephone Number  
 Source : HOME\_PHONE  
 Min/Max: 0/1  
 Existence: Optional  
 Class: Critical  
 Indexes: None

staff - A flag that indicates if the person is a staff member.

OID:  
 Syntax: Boolean  
 Source : STAFF  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: None

alumni - A flag that indicates if the person is an alumni member.

OID:  
 Syntax: Boolean  
 Source: ALUMNI  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: None

faculty - A flag that indicates if the person is a faculty member

OID:  
 Syntax: Boolean  
 Source: FACULTY  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: None

student - A flag that indicates if the person is an alumni

OID:  
 Syntax: Boolean  
 Source: ALUMNI  
 Min/Max: 1/1  
 Existence: Required  
 Class: Normal  
 Indexes: None

dateOfBirth - The person's birth date.

OID:	
Syntax:	Generalized Time
Source :	BIRTH_DATE
Min/Max:	0/1
Existence:	Optional
Class:	Critical
Indexes:	Order

gender - A code that is used to identify the person's gender.

OID:	
Syntax:	String
Source :	GENDER
Min/Max:	0/1
Existence:	Optional
Class:	Critical
Indexes:	None

### **Lessons Learned/Problems**

There were little problems encountered with this project other than the initial data gathering process and creating a template of what our University would need with a “white pages” directory. We used saved web links to return to previously visited example pages and took extensive notes when reviewing the associated RFC documentation to ensure we could return to the information we gathered if additional specificity was needed.

### **Final Results/Recommendations**

It is the contention of this group that the LDAP project is a great way to learn some management infrastructure in the IT realm. It was important for us, on this project, to abstract our goals enough where the suggested structure would be easily implemented at smaller and larger institutions. This helped us increase the management as well as technical side of our systems administration capabilities.

**References**

- "eduPerson Object class version 1.0". 22 January 2001. 16 April 2003  
<<http://www.georgetown.edu/giia/internet2/ldap-recipe/eduPerson-schema.ldif>>
- "Introduction to ASN.1". ASN.1 Information Site. 8 August 2002. 16 April 2003  
<<http://asn1.elibel.tm.fr/en/introduction/index.htm>>
- Barton, Thomas. "Practices in Directory Groups". 29 January 2002. 16 April 2003  
<[http://middleware.internet2.edu/dir/groups/rpr-nmi-edit-mace\\_dir-groups\\_best\\_practices-1.0.html](http://middleware.internet2.edu/dir/groups/rpr-nmi-edit-mace_dir-groups_best_practices-1.0.html)>
- Gettes, Michael R. "A Recipe for Configuring and Operating LDAP Directories". 10 October 2002. 16 April 2003 <<http://www.georgetown.edu/giia/internet2/ldap-recipe/#DIT>>
- Hodges, Jeff. "Introduction to Directories and the Lightweight Directory Access Protocol". 18 June 1997. 16 April 2003. <<http://www.stanford.edu/~hodges/talks/mactivity.ldap.97/index2.html>>
- Wahl, M., T. Howes and S. Kille. "Request for Comments: 2251 - Lightweight Directory Access Protocol (v3)". Network Working Group. December 1997. 16 April 2003  
<<http://www.ietf.org/rfc/rfc2251.txt>>

**Project:** Bush Hog, LLC System Administration Practices

**Author:** C. Judith Nyabando

---

### **Introduction**

The purpose of this project was to get exposed to real world system administration and to analyze system administration practices at Bush Hog, LLC located in Jonesborough, Tennessee. This paper will discuss Bush Hog's system infrastructure, server and desktop management, help desk management, change management, security issues, and back up policies. The paper will also point out the good aspects of the practice and areas of improvement or alternative solutions in relation to issues discussed in class and in *The Practice of System and Network Administration* textbook.

### **Bush Hog's Background**

Bush hog, LLC is a farm equipment manufacturing company based in Selma, Alabama and has branches in other parts of the country including Jonesborough, Tennessee. The Jonesborough branch has about 200 employees.

### **System infrastructure**

The Bush Hog, Jonesborough has two main networks: Production network which is used for production purposes and the regular network used for general purposes. The production network provides services to the production equipment. The production equipment includes equipment for cutting metal to produce a variety of shapes and equipment for painting finished products. Administration of this network is outsourced to a company in Knoxville that has expertise in networks of this type. The regular network is managed by Bush Hog system administrators. The regular network has 4 HP net servers, 1 main frame print server controller, 113 PCs, and 24 printers.

### **IT Staff**

Bush Hog main office in Selma, Alabama has about four system administrators. The Jonesborough location has two system administrators. System administration falls under the MIS department. The MIS department has three areas of responsibility: system administration, telecommunications administration, and Bush Hog security monitoring. Telecommunications administration involves configuring the phone lines to meet the requirements of the customers at Bush Hog for example configuring voice mail or configuring the executives office phones to ring both in the executives' office and their secretaries' office. Security monitoring involves assessing data feed from security cameras on Bush Hog premises and monitoring automatic locking and unlocking of gates and doors on the premises. The two system administrators are responsible for the majority of these tasks.

### **Servers and Services**

Bush Hog, LLC rents a server space on an IBM 3970 mainframe server in Atlanta. The server runs their ERP software which includes production schedules, shipping, inventory, and purchasing. As mentioned above Bush Hog, Jonesborough has four servers. The servers are housed in a lockup closet in the MIS office. Each of these servers is dedicated to providing a single service. These servers are file server, Cyber Docs and Kronos server, SDRC Server,

Email server, and the print controller. The Cyber Docs maintains drawing used by the production and engineering departments. Kronos manages the time card system. When employees come to work and leave work they swipe their cards on the card reader and Kronos logs their start time and end time, respectively. The information is then used to run payroll. SDRC is for CAD drawings. The email server is a Microsoft Exchange server which replaced Group Wise. All these services run on Windows 2000 server. The print job controller works with the mainframe server in Atlanta to manage print jobs like drawings. The web service is provided from Alabama, on a corporate level. Some of their printers are network printers and some of them are desk top printers. There is one main network printer in the administration offices. The printer functions as a copier and fax machine as well.

### **Desktops**

Most of the desktops are Hewlet Packard, some are Compaq and some are Sony. These desktops come from the vendor with pre-installed Operating System (OS) and the system administrators have an image for each type which they use to rebuild the systems if the need arises. All desktops run Windows XP.

### **Network Configuration**

DHCP is used for IP address configuration for most of the machines except for servers, network printers, and desktops in the manufacturing plant which have static IP addresses. They have an electronic and hardcopy spreadsheet that has a record of every machine, its model number, serial number, IP address and customer assigned to the machine:

<b>Item</b>	<b>Model #</b>	<b>Serial #</b>	<b>IP Address</b>	<b>Assigned To</b>
HP DeskJet 1220Cse InkJet Printer	DC500		10.1.1.168	Purchasing
Sony Digital PC	RX650	6436384	DHCP	Mary T. Ruth

### **Help Desk Management**

Since the system administration team is small and there are a small number of systems to manage, they do not use any ticket tracking software. However, they use email in collaboration with task manager which is part of Outlook to keep track of customer needs. When a customer sends them an e-mail they log it in the task manager. They also have handhelds which they use to record tasks when they are away from their desks. Once they return to their desks they log those tasks to the task manager. They also have a telephone number that customers can call to report problems. The downside of the telephone system is they have to manually log requests.

The team is also highly visible to their customers. Indeed, it is difficult to hide from 100 people, but the attitude of the system administrators can drive customers away. The environment at Bush Hog appeared customer-friendly.

### **Change Management and Maintenance Windows**

Most of the employees in the administration offices work from 8am to 5pm, therefore maintenance windows are scheduled after 5pm.

### **Transition from GroupWise to Microsoft Exchange**

Bush Hog, LLC recently converted from GroupWise to Exchange. The transition was transparent to the customers as far as namespaces are concerned. Their e-mail addresses remained the same. Also, they still have access to their mail from the GroupWise until a certain date. This will give customers enough time to transfer important e-mails to Exchange. The system administrators did not have enough knowledge about Exchange so the company sent them for training.

### **VNC**

Virtual Network Computing (VNC) is a free software that the Bush Hog MIS team use to manage desktops from their own desktops by entering the host name or IP address of the machine they want to access. They use the tool for remote installation of software on the desktop, among other things. This tool allows them to work on several machines at the same time from their desktops. The downside of this tool is that the administrators cannot work on the remote machine if someone else is using it. If they take control of the computer no one else can use it at that time. They can also use this tool to access servers in Alabama.

### **Naming**

There is no naming policy on a corporate level therefore system administrators at branch level have the right to decide namespaces. The Bush Hog, Jonesborough server names consist of TN + a number where TN indicates Tennessee. The only problem with this is there might be naming conflicts if another branch is opened in Tennessee. Usernames are first name + last name initial. However, conflicts have occurred so they are now assigning first name initial + last name usernames. The usernames have to be at least six characters.

### **Backup and Security Policies**

Back ups are done every night on DLT 80 tapes. These back ups are full back ups. It seems the main security goal for Bush Hog, Jonesborough is to restrict access to resources from unauthorized individuals. Security is applied on a group level and individual account level. Each department is a group and only members of a particular group have access to the resources of that group. When an employee leaves the company, access to company resources is disabled immediately and the proximity card is also collected for they are reusable.

### **Conclusions**

Below are conclusions drawn from this study. They are presented in two categories: good aspects and recommendations. Overall, the current practices are effective in meeting the company's needs. In the long run some areas may need to be adjusted to meet growth needs.

### **Good Aspects**

Given the fact that Bush Hog, Jonesborough has been in existence for about two years, the practice of system administration at this location is good. The practice is not ad hoc even though it is not advanced. There is definitely a defined process that is followed by the system administration. For instance, every machine and its properties at the location are documented on a spread sheet as mentioned above. This serves as a reference document to the system administrators and others in need of the information. There is also a namespace policy, and customer request are documented. The documentation of these aspects makes it easier to adjust

to growth. The use of automation and remote access tools such as VNC are also good system administration practices.

### **Recommendations**

The main recommendations in this section are in regards to economics. The current process definitely works efficiently but as the company grows changes will need to be done, including economics planning.

### **Printers**

The administrative wing at Bush Hog, Jonesborough is located in one area on the same floor but they have 24 network and desktop printers which may not be economical. Now these printers include printers for other departments like Welding not located in the administrative area. The main network printer is located in a central area within almost 50 feet from the furthest desk in the administrative wing. Some people have desktop printers even though the network printer room is right behind them. This may not be necessary unless the information is very confidential. On the other hand the current set up may not be an economical concern for Bush Hog.

### **Backups**

A full backup every night may not be necessary. Their main reason for implementing a full backup process is that if there is a need to backup they will only do it from one tape. The cost of backing up may not be significant at this point but in the long run as the company grows the cost of full backups will grow. It is not clear how often the tapes are recycled and how critical are the data being backed up. However, the following backup process might be more economical:

- do full back up once a week
- do incremental backups everyday
- recycle the incremental backup tapes

### **References**

Limoncelli, Thomas A. and Christine Hogan. *The Practice of System and Network Administration*. Addison Wesley. 2001.

Rice, Mark. Technical Services Manager. Bush Hog, LLC, Jonesborough, TN.

# *Lisa Write-up*

## **Abstract**

The LISA conferences have produced volumes of articles on a diversity of topics. The papers result in a collection of experiences, methods, theories and ideas regarding system administration. We surveyed and analyzed this collection of work. Our objective is to discuss the major trends found in the articles presented at LISA conferences from 1993 to 2001. Also we provide some comments based on our findings.

## **Introduction**

Papers presented at the LISA Conference during the years 1993-2001 reflect the issues faced by system administrators over the years. It is instructive to consider the state of computing during these years. In 1993, the first LISA conference that we are studying, the trend in computing was shifting from mainframes to Unix workstations. Towards the late 90's the influence of Intel based PCs was beginning to be felt. Corresponding to the growth of the PC came an exponential growth in Internet use for businesses, universities, and home users everywhere. Toward the end of the articles reviewed here, a major move is seen towards Linux, and this is reflected in the topics selected.

The articles presented represent what problems system administrators were trying to solve given the historical milieu in which they were working. Some problems seem to be universal no matter what else was going on in the world of computers. Other topics had their heyday early, and gradually lost their importance as new technologies were developed to render that particular subject no longer a concern. Some topics made their appearance relatively late in the LISA lifecycle, but are the concern of many papers once they emerge. Other articles cover topics that appear only once. We only cover the keywords that had more than ten articles. Following is a discussion of previous work, our analysis of the most popular keywords presented at LISA, and some comments based on our analysis.

## **Previous Work**

At the 1999 LISA conference, Eric Anderson and Dave Patterson of the University of California at Berkeley presented a paper entitled "A Retrospective on Twelve Years of LISA

Proceedings". This article presents a comprehensive view of topic frequencies over the first 12 years of the LISA conference. Anderson and Patterson offer no real explanations of why trends develop. Our paper differs in that we are looking at only a subset of the trends, and trying to put them into perspective with what was happening in the larger Information Technology field.

### **Methodology**

Appendices 1 and 2 were developed by reading the abstracts for all papers presented at the LISA conference from 1993 until 2001. Only papers that were freely available online were considered in this study. The papers were divided evenly into 3 groups, and one team member determined the topic keyword for each article assigned. Several meetings were conducted to determine an agreed upon set of keywords to use in this study; these are listed in Appendix 1.

After all the articles had been processed, one team member compiled all of the results into a master spreadsheet. The analysis consisted of sorting the spreadsheet by keyword and then by year to produce Appendix 2. Appendix 2 was used to discover overall trends in the paper topics. We were able to see not only how many times a topic had been covered by a LISA paper, we were also able to see the years that the topic occurred. The team then got together to produce the first draft. Individual members then took turns refining the document until it reached its final form. The Appendix 3 lists the articles along with the year presented, keyword assigned, title, and authors.

The individual team members spend a combined 31 hours in analyzing the articles to produce keywords. A further 9 hours was spent combining and refining the individual results. 12 hours were spent in meetings before and during the creation of the first draft. A combined 27 hours was spent on producing the yearly summaries. The team members spent a final 6 hours in working through version of the first draft to produce the final product. In total, we spend 85 hours on this project.

### **Automated Configuration Management**

The keyword with the largest number of articles was automated configuration management. With a large and always growing number of hosts to manage, system administrators have always found it important to automate the process of software configuration. The general strategy for most of the solutions seemed to be abstracting the process of software configuration so that system administrators would not have to process the tedium of configuring

every single service in detail all the time. Papers coming out of automated configuration management have grown somewhat as time passed, but not noticeably so. This probably corresponds to the growth in use of computing in the late 1990's, but the small amount of growth is surprising. Two major categories that fell out of the analysis were infrastructure systems and articles using the cfengine package. Also a few other articles are described.

Eleven articles describe solutions that provide infrastructures for software deployment and configuration. These involved setting up a centralized database to keep track of management, servers to deploy software to client machines, and methods for clients to automatically configure the software once deployed. All systems mentioned logging and tracking of client configurations. The Aurora system ("Morgan Stanley's Aurora System: Designing a Next Generation Global Production Unix Environment"), a system developed at Hewlett-Packard ("Automating Infrastructure Composition for Internet Services") and Athena ("Anatomy of an Athena Workstation") are all infrastructures that allow users to access programs and data from any workstation. These all describe systems that took enormous efforts to create. It seems likely that if available, such systems would become more widely used as computing becomes more large-scale.

The software package cfengine has several articles describing various aspects of its usage. Mark Burgess, the author of cfengine, started the project in 1993 (<http://www.cfengine.org/cfdetails.html>). Five articles cite the use of cfengine, more than any other software package. This number indicates that it is in use by system administrators running large installations. One article is a general discussion of its usage ("Use of Cfengine for Automated, Multi-Platform Software and Patch Distribution "), and another ("CfAdmin: A User Interface for cfengine") is a description of cfadmin, a user interface that tries to make the use of cfengine easier. The article "Adaptive Locks for Frequently Scheduled Tasks with Unpredictable Runtimes" describes a scheduling technique that is implemented with cfengine. One article ("File Distribution Efficiencies: cfengine Versus rsync") compares the effectiveness of cfengine with that of rsync for transferring files. The last article, "TemplateTree II: The Post-Installation Setup Tool," describes a configuration management system that is built on top of cfengine.

There were also several other articles covering a range of topics. One of note is UNIX-specific ("An Analysis of UNIX System Configuration"). Another describes a system written in Java that is used to keep client programs configured correctly ("System Management With

NetScript"). Yet another describes a freely available software package called Modules that in wide use that keeps track of installations and configurations ("Abstract Yourself With Modules"). The last paper of note is a description of scheduling installation and configuration tasks ("Scheduling Partially Ordered Events in a Randomised Framework: Empirical Results and Implications for Automatic Configuration Management"). It suggests a method that would streamline the process of configuration management. This is an excellent direction to follow, and we hope that future LISA automated configuration management papers will follow.

### **Software Installation and Updates**

Software installation and updates was the second most popular topic present at LISA conferences. Although several papers with this keyword have a lot to do with automated configuration management, these articles concentrate only on the installation of software and updates. Similar to automated configuration management, the articles here did not grow noticeably over the years. This task has been represented evenly throughout the years of our analysis. The papers fell into three categories: large-scale systems, UNIX operating system installation, and patch management.

The bulk of the articles dealing with software installation and updates describe large-scale systems that automate these two processes. Most of these are cross-platform and include version management. Generally they implement a centralized database that tracks client configurations. In addition, two added the capabilities to customize client computers ("Managing and Distributing Application Software" and "Soft: A Software Environment Abstraction Mechanism"). "Automating Request-based Software Distribution," describes a system that includes authorization in the distribution of software. "A Management System for Network-Shareable Locally Installed Software: Merging RPM and the Depot Scheme Under Solaris" describes merging RPM, a popular packaging tool, with their software distribution methods. The last article, "Depot-Lite: A Mechanism for Managing Software," describes making the process of software installation easier to use. The authors' motivation for this is to lessen the amount of expertise needed to perform these tasks, a welcome direction for future solutions.

A few of the articles specifically cover the installation of UNIX. The first, "A Linux Appliance Construction Set," discusses a few configurations of Linux in depth. The hope is that this coverage will aid system administrators from having to duplicate the author's efforts. The

second two articles, "Automated Installation of Linux Systems Using YaST" and "Automating Dual Boot (Linux and NT) Installations", describe tools that automate the installation of operating systems; these could be useful for system administrators in saving time to reinstall operating systems. The last article, "Enterprise Rollouts with JumpStart", describes a similar system for Solaris machines, but is interesting in that it has discussions on how to optimize the time taken for installation.

Two articles focused on patch management. The first article ("Patch Control Mechanism for Large Scale Software") describes a system that successfully manages the installation and application of patches. The second article ("Cro-Magnon: A Patch Hunter-Gatherer") describes a system that accomplishes two tasks. The first is to automatically check the Internet for any new patches that an administrator specifies, and the second is to allow the easy application of the patches to target machines. As the author of the Cro-Magnon paper notes, "Although we presume that most large networks use homegrown automated tools to provide many of the functions included in Cro-Magnon, little has been published about such systems." We hope to see more papers directed towards the task of patch management in the future.

## **Security**

There have been 16 articles on security presented at LISA, evenly distributed between 1995 and 2000. Many of these articles discuss tools that System Administrators have build to allow them to automate some security tasks. Examples of this type of article are "Multi-platform Interrogation and Reporting with Rscan" (1995), "TITAN" (1998) and "NOOSE – Networked Object-Oriented Security Examiner" (2000). The topics discussed in the security articles reflect the computing environment in which the LISA presenters worked. Thus, early articles tended to focus on the threats from internal sources. As the Internet continued to grow, more articles focused on the threat from outside hackers. The first articles to address this outside concern were presented in 1999, and included such articles as "Snort – Lightweight Intrusion Detection for Networks" and "Managing Security in Dynamic Networks".

Another security topic that has appeared throughout the proceedings is how to control privileged access to systems. Examples of this type of article are "Exu- A System for Secure Delegation of Authority on an Insecure Network" (1995) and "SSU: Extending SSH for Secure Root Access" (1998). There was only one mail security article presented in our target

years. “ssmail: Opportunistic Encryption in sendmail” was presented in 1999. As mail use becomes more and more important, we expect to see more articles on this topic. “Moat: A Virtual Private Network Appliance and Services Platform” (1999) details another trend in Networks, virtual private networks. This is the only LISA article so far to discuss this topic, but we expect to see more on this at future LISA conferences.

One of the most interesting articles was “Analyzing Distributed Denial of Service Tools: The Shaft Case” from 2000. This article takes a close look at a tool that hackers can use to bring down a server. By looking at how the attack is accomplished, it is hoped that its use can be prevented. This article, more than others, points out new threats being made against systems.

### **Network Administration**

There have been 15 articles on Network Administration presented at LISA during our coverage years. Most administration articles have appeared in the last few years. The rise in the number of articles correlates directly to the increase in network performance due to the Internet.

Several articles such as “Wide Area Network Ecology” (1998) and “ND: A Comprehensive Network Administration and Analysis Tool” (2000) specifically mention that their development was in response to increase Internet Traffic. Many other articles describe tools for analyzing network traffic such as “MRTG – The Multi Router Traffic Grapher” (1998), “FlowScan: A Network Traffic Flow Reporting and Visualization Tool” (2000) and “The CoralReef Software Suite as a Tool for System and Network Administrators” (2001).

Two articles present tools for using the SNMP protocol more effectively: “Thresh – A Data-Directed SNMP Threshold Poller” (2000) and “Specific Simple Network Management Tools” (2001).

### **Mail Management**

There have been 11 articles about Mail Management; most of them evenly split between the years 1996 and 1998. This clustering corresponds to the rapid growth of and increasing importance of email during the second half of the nineties. The topic most vexing Mail

Administrators during this period must have been dealing with mailing lists. Six of the 11 articles deal with mail list management, from “MajorCool: A Web Interface to Majordomo” (1996) to “Dynamic Sublist: Scaling Unmoderated Mailing Lists” (2001).

The other hot topic in Mail Management was performance. The first performance article “How to Get There From Here: Scaling the Enterprise-Wide Mail Infrastructure” (1996) clearly reflects the increasing need for mail services. Other articles such as “Automatic and Reliable Elimination of E-mail Loops Based on Statistical Analysis” offer some concrete steps to take to make mail perform better.

### **Account Management**

Our analysis resulted in 10 articles focusing on Account Management. The articles range from 1994 to 2000. The articles were evenly distributed through these seven years. Most of the efforts concentrated on both centralizing and standardizing account management over multiple platforms and hosts. The primary motivation for centralization is the workload involved in account management; creating, removing, changing and maintaining large numbers of accounts for different hosts consume a lot of time. As a result the general idea we found was to abstract idea of an account into a centralized system. Administrators would then work with this centralized system to work with the account. The authors generally worked with integrating several systems including Windows, UNIX, Macintosh, Novell, and VMS. Also login and directory standards such as NIS and LDAP were sometimes employed to aid the centralization process. All of the articles ran into problems dealing with the creation of the centralized account management tools. The biggest of these were programs that were not flexible enough to support the systems being built; several of the systems had to create workarounds for such situations.

Other than centralization and standardization, a few articles focused on specific aspects of account management. These were implementing a single login system ("THE BNR Standard Login (A Login Configuration Manager)"), password management ("SPM: System for Password Management"), and permissions management ("Priv: Secure and Flexible Privileged Access Dissemination" and “Implementing Execution Controls in Unix”).

Overall the area of account management seems to focus on managing a large number of accounts over a diverse number of hosts. The steady distribution of articles is surprising; it seems as if that the number of articles submitted did not increase with the growth of desktop computing in the late 90's. Even with these results, we suspect that this area can only become more important as computer use grows.

### **Comments**

The Internet era has brought both boom and bust to the economy. As the economy rose higher and higher during the heyday of the Internet several companies found their way into high technology areas. However, the growth eventually stopped and years later we are still dealing with the results of the inflation. For businesses this means growth, merging, splitting and other major changes. Some of the papers that we analyzed mentioned outsourcing. Such issues bring a myriad of problems and challenges to system administrators everywhere. These papers study issues like how to split networks, how to manage remote offices, and how to integrate heterogeneous systems. We can only predict that the prevalence of these kinds of papers will increase over time until the economy becomes more stable.

A change that has surprisingly not found its way into topics at LISA is the rise of the World Wide Web service. The web, like e-mail, has become more and more important to the everyday operations of businesses. We would have thought that this would be reflected in more articles about managing web resources. However, only a few LISA articles have addressed this trend: one in 1998 on administrating distributed Web servers and one in 2001 on a Web-based application server. We conjecture that the system administrators who deal with web services must be attending more specialized conferences than LISA. We also feel that this is an area that should be addressed in future LISAs.

Two major issues arise in the area of networking. A large general trend in computing will be the pending switch to IPV6. Such a major switch will be necessary for almost all organizations connected to the Internet. This will result in replacing applications that are not compatible with IPV6, and will result in huge efforts for both developers and system administrators. Another trend in computing seems to be towards mobile computing. Small devices like personal digital assistants and cell phones have the ability to read mail and to surf the web. These devices also use wireless technologies to communicate with the Internet.

Networking such devices with wireless technologies can only result in more problems in terms of security and management for system administrators.

What remains is a discussion of other miscellaneous topics. System administration is not very formalized as a profession, and the few number of papers on this issue reflects this notion. One describes the evaluation of system administrators (“The System Administration Maturity Model – SAMM”). There are also a few papers on policies. However, we hope to see a growth in these topics as the administration of computing resources can only grow larger in the future. Also, we were surprised to see no papers on self-healing, a topic that seems to be on the rise in computing. Last, with the growth of computers mission critical systems will inevitably grow. Here we hope that more works will be presented on 24x7 availability, reliability and robustness in future LISA conferences.

### **Conclusion**

Our overview includes a few of the key topics presented in past LISA conferences. New technologies will soon be deployed out into businesses, universities and other organizations. These include wireless technologies, video/voice over IP, IPV6, Internet2, and a larger installed base for broadband. Security will continue to be a hot topic. Such improvements can only result in huge changes to the technology arena, and therefore in a corresponding growth of the system administration field.

## Appendix 1 – Keywords Used

Account Management  
Automated Configuration Management  
Availability  
Backup  
Change Management  
Clustering  
Collaboration  
Computer Lab Administration  
Data Visualization  
Data Warehousing  
Database Management  
Dependency Management  
Directory Services  
Distributed Administration  
Distributed Filesystems  
Domain-Specific Application  
Filesystem Management  
Hardware Configuration Management  
Help Desk  
License Management  
LISA Analysis  
Log Management  
Mail Management  
Metrics  
Mirroring  
Mission Critical Systems  
Monitoring  
Multicasting  
Network Administration  
Network Configuration  
Network Performance  
Network Topology  
Outsourcing  
Personnel Management  
Phone Systems  
Physical Relocation  
Policies  
Printer Administration  
Process  
Remote Administration  
Restructuring  
Scalability  
Scripting  
Security  
Self-healing

Software Evaluation Methods  
Software Installation and Updates  
System Administrator Evaluation  
System Performance  
System Recovery  
Training  
Unix Kernel Programming  
Usenet Server Administration  
User Account Customization  
User Simulation  
Version Management  
Work Tracking

## Appendix 2 – Keywords ranked by number of articles

Automated Configuration Management	24
Software Installation and Updates	21
Security	16
Network Administration	16
Monitoring	13
Mail Management	11
Account Management	11
Process	8
Distributed Administration	7
Distributed Filesystems	6
Network Topology	6
Work Tracking	6
Backup	5
Filesystem Management	5
Network Configuration	5
Printer Administration	5
Log Management	4
Personnel Management	4
Restructuring	4
System Performance	4
Usenet Administration	4
Server	4
Availability	3
Collaboration	3
Computer Administration	3
Lab	3
Database Management	3
Directory Services	3
Policies	3
Scripting	3
Training	3
Change Management	2
Clustering	2
Data Visualization	2
Domain-Specific Application	2
Mission Critical Systems	2

Scalability	2
System Recovery	2
Data Warehousing	1
Dependency Management	1
Hardware Configuration Management	1
Help Desk	1
License Management	1
LISA Analysis	1
Metrics	1
Mirroring	1
Multicasting	1
Network Performance	1
Outsourcing	1
Phone Systems	1
Physical Relocation	1
Remote Administration	1
Self-Healing	1
Software Evaluation Methods	1
System Administrator Evaluation	1
Unix Kernel Programming	1
User Account Customization	1
User Simulation	1
Version Management	1