Inessential Linux-Athena

The Student Information Processing Board
Derek Atkins
Emil Sit
Aaron Ucko
Salvatore Valente

August 24, 1998

1 Introduction

Congratulations! You are now reading the Inessential Guide to using the Linux operating system[1] to access all of your favorite Athena services over the MIT network and the Resnet extension.

This guide assumes that you are using RedHat-Athena. However, it should also apply with some changes to other Linux distributions. Note in particular that SIPB no longer supports Slackware; we recommend that you seriously consider switching to RedHat.

1.1 ...But I'm Not Using RedHat-Athena

Installing RedHat-Athena is easy; detailed instructions appear in "Installing RedHat-Athena Linux on MITnet." If you are running Slackware, any other non-RedHat-based distribution,[2] or an old version of RedHat (older than 2.0), you will have to wipe out your old installation and install RedHat from scratch. Before doing so, you should back up important files lest you lose them. These include the contents of `/etc` and any personal files you may have on your machine. This procedure can be a nuisance, especially if you have made a lot of random customizations to your machine, but SIPB strongly recommends it anyway, particularly if you have not installed the Linux-Athena software; again, RedHat is the only distribution SIPB supports.

Before installing, you should know your complete hardware configuration, as you will be asked for it. If you cannot determine this, here are some tips for identifying unknown hardware: A really cheap network card is likely an NE2000. If you don't know what kind of video card you have, a program called `SuperProbe` may tell you.

We should also warn you that new hardware may not be supported; as a general rule, if the DOS or Windows version of the driver is still being debugged, the device probably will not work with Linux; if your hardware is not listed, this may be why. You may be able to use a driver for simpler versions of your hardware; for instance, the generic VGA X server works with almost all video cards.

If you are using "vanilla" RedHat 4.2, you can upgrade to RedHat-Athena very easily. Just `su` to or log in as root, and type the following commands:

```
mkdir -p /sipb-nfs/redhat
mount sipb-nfs:/redhat /sipb-nfs/redhat
rpm -ivh /sipb-nfs/redhat/4.2/i386/RedHat/sets/Athena/*
```

2 Assumptions

This paper assumes that you have already successfully installed RedHat-Athena on your computer.[3] You should be able to log in as root and run simple commands. For example, you should be able to use the `passwd` command to change the root password on your computer to be something that nobody else knows.[4] You will **NOT**, of course, tell **anyone**.[5]

This paper also assumes that you are familiar with the basic concepts of Unix. For example, you should know what a "dotfile" is, you should know how to change your "path", and you should have some idea what a "pipe" is. You should also know how to navigate around your filesystem and how to copy and move files. It might also be helpful if you were familiar with standard Unix file permissions and ownership. If you are not familiar with these terms, then it's a good idea for you to find a book on Unix for beginners; you could also read some of Information Systems' documentation (particularly "Managing Your Athena Account") or other SIPB documentation (particularly "Inessential Athena"), or attend some introductory Athena minicourses.

[1] If you don't know what Linux is, and you would like to find out, a very good source of documentation is the WWW page `http://www.linux.org/`.

[2] If you are using a recent version of Debian, you may be able to follow the instructions for RedHat 4.2 with some modifications; `dpkg` supports installing RedHat packages now.

[3] SIPB offers a brief paper called Installing RedHat-Athena Linux on MITnet. You can pick it up in W20-557.

[4] If you change your root password you'll have to propagate your changes manually to `/etc/passwd.local` because `/etc/passwd` is reset on boot.

[5] Even if you are tortured.

2.1 For More Information...

There are many useful sources for information about Linux.

- O'Reilly and Associates offers a number of useful books related to maintaining and running variants of Unix, and several books specifically on Linux. Some of these books are available at the Coop.
- For general information on configuring a host for use on the net, you should consult the "`NET-2-HOWTO`" in `/mit/linux/docs/HOWTO`. They may also be installed on your system in `/usr/doc/HOWTO`. Basic configuration for MITnet is covered in the SIPB Linux Net Installation ("Installing RedHat-Athena Linux on MITnet") document. If you need help after reading these files, call or drop by the SIPB office (W20-557, x3-7788) and we will try to help.
- The Usenet newsgroups in the `comp.os.linux.*` hierarchy discuss setting up, configuring and using the Linux operating system. You may want to read the newsgroups for a while to get a feel for what problems other users encounter.[6]
- MIT has four mailing lists concerning Linux, all of which are archived in `discuss`:[7]

List name	Discuss path
`linux-help`	`charon.mit.edu:/usr/spool/discuss/linux-help`
`linux-announce`	`charon.mit.edu:/usr/spool/discuss/linux-athena`
`linux-dev`	`charon.mit.edu:/usr/spool/discuss/linux-dev`
`linux-afs`	`menelaus.mit.edu:/usr/spool/discuss/linux-afs`

  Please type `blanche linux-announce -a $USER` to add yourself to the `linux-announce` mailing list. You should probably also consider reading `linux-help` because other users may have the same questions you do.
- There are many WWW pages concerning Linux. Two useful ones are given by the URLs `http://web.mit.edu/linux/www/` and `http://www.linux.org/`
- Man pages are invaluable in gaining an understanding of how commands work and how to use them properly. You can read man pages by using the `man` command. A typical usage would be `man tar` to get help on the usage for "`tar`".

3 Kernels

RedHat-Athena 4.2 is based on RedHat Linux release 4.2, the fourth major release of the RedHat Linux distribution. This distribution is based on version 2.0.32 of the Linux kernel. (A kernel is the heart of an operating system, providing basic functionality used by other components. A distribution includes a kernel and the rest of the software forming the operating system.) The latest kernel version that works with Linux-AFS is 2.0.35; although version 2.0.32 also works fine, you might want to consider upgrading anyway because version 2.0.35 fixes some problems present in version 2.0.32. Version 2.1.n kernels will *not* be supported; these kernels are experimental and unsupported by SIPB.

If the kernel you are using does everything you need it to do, you should be able to keep using it. However, you might want to install another kernel, either because you want to use hardware (a sound card, for instance) not supported by the kernel you currently have installed or because you would like to upgrade to a newer supported kernel.

Linux-AFS is very dependent on kernel versions, so if you upgrade your kernel you will also probably need to upgrade Linux-AFS. See `/mit/linux/packages/Linux-AFS/README` for more details. If you upgrade to a version 2.0.n kernel from a 1.m.n kernel, you will have to make a lot of other changes; see `/mit/linux/kernel/linux-2.0/Changes` for more details. If you are using RedHat Linux 3.0 or older and want to use a 2.0.n kernel, it is probably easiest just to upgrade to RedHat-Athena 4.2.

To actually build a kernel, you should first ensure that the version you want to install is in `/usr/src/linux`. If it isn't, you can obtain a kernel from `/mit/linux/afskern`; all the kernels in this directory are stable and

[6] SIPB has an Inessential NetNews document, if you want more information on reading news.
[7] If you are unfamiliar with `discuss`, you can read "Inessential Discuss," available in W20-557.

should work with AFS. Before you can compile a kernel, you must first unpack it by running the following commands as root:

```
cd /usr/src
zcat /mit/linux/afskern/linux-version.tar.gz | tar xvf -
```

where *version* is the version you want to build.

To build the new kernel, you can now run the following commands as root:

```
cd /usr/src/linux
make config
make dep
make zImage
```

After you run these commands, your new kernel will be the file `/usr/src/linux/arch/i386/boot/zImage`. You can test this kernel by putting it on a floppy: put a blank disk in the drive which DOS would call `A:` and type (as root)

```
cd /usr/src/linux
make zdisk
```

If the new kernel works, you can install it with the commands

```
cd /usr/src/linux
make install
```

If you are building a version 2.0 or newer kernel, you can run `make xconfig` or `make menuconfig` instead of `make config` if you want a friendlier interface. When configuring the kernel, be sure to enable module support and to "set version information on all symbols for modules"; otherwise, Linux-AFS will not run. If you have configured any drivers as modules, you will have to add some steps to the above process; type `make modules` after compiling the kernel, and `make modules_install` after installing it. More information on building kernels is available in `/mit/linux/docs/HOWTO/Kernel-HOWTO`.

4 Installing Additional Packages

If there's a package you would like to use but failed to install initially, the situation is easy to remedy. Just NFS-mount `sipb-nfs` by typing `mount sipb-nfs:/redhat /sipb-nfs/redhat` as root and run `glint`. When `glint` starts, click on "Configure," enter `/sipb-nfs/redhat/4.2/i386/RedHat/RPMS`, and click on "Save." If you then click on "Available," `glint` will allow you to select more packages to install. If you are unsure whether a package is already installed, `glint` will tell you. (Installed packages are listed in the first window `glint` shows.) Another way to determine which packages are installed is to type `rpm -qa`. (This will produce a long list, so you'll probably want to save it to a file.)

4.1 Updates

RedHat-Athena will be updated occasionally with fixes to problems which have been found in it. To install updated versions of packages you have installed, attach the `linux` locker and then run the script

```
/mit/linux/redhat/current/i386/updates/update.pl
```

as root. This script will update only packages which you have already installed and which need to be updated, and will attempt to keep Athena-specific customizations. If you are unsure whether there have been recent updates, run the script anway; if there have been, it will install them, and if there haven't, it will tell you.

4.2 Missing Workstation Software

You will not be able to run all of the software you can run on public workstations on your computer because many lockers do not have support for Linux binaries. Framemaker, for instance, is not currently available. As Derek once said,

> The SIPB has been capable of convincing many organizations to support Linux; however, we can only encourage them, we cannot force or coerce anyone to do anything. Generally we have encouraged groups by donating the disk space to hold the Linux binaries. FYI: The third party software like Maple, Matlab, and Xess is only available because those companies have seen fit to make their products available under Linux.

5 Customizing Your Machine

By default, your machine acts like any private Athena workstation: among other things, this means that few users can telnet to it, that there are no local mail addresses, and that it mounts your home directory by AFS. You may want to change various aspects of these.

5.1 User access control

User information for Unix machines is stored in a file called `/etc/passwd`. By default, your system will allow any valid Athena user to log in when physically at your machine, but only users in this file to log in remotely (i.e., by telnet or rlogin). In order to allow users the ability to log in remotely, you need to put them in the `passwd` file.

To do this, you need to become root and run

```
/bin/athena/hesinfo user passwd >> /etc/passwd.local
/bin/athena/hesinfo user passwd >> /etc/passwd
```

This will append the appropriate user information to the file. Note that by default even you aren't in the file, so you'll probably want to add yourself ASAP.

You may be wondering why you should change `/etc/passwd.local` in addition to `/etc/passwd`. The reason is simple: Any changes you make only to `/etc/passwd` will be lost because `/etc/passwd.local` is copied to `/etc/passwd` every time you boot your machine. (There are several reasons for this, such as that it keeps `/etc/passwd` from growing too large and allows for improved security.) The same applies to `/etc/group` and `/etc/group.local`.

If you wish to restrict access of users of the machine to just those in the password database, edit the file `/etc/athena/rc.conf` and change the line `NOCREATE=false` to `NOCREATE=true`. Then run `touch /etc/nocreate`. This will only allow users in `/etc/passwd.local` to log in to the machine whether they try to log in remotely or at the console. If you want to change this back, remove `/etc/nocreate` and change `/etc/athena/rc.conf` to say `NOCREATE=false` instead of `NOCREATE=true`.

At the other extreme, it is also possible to let everybody log in to your machine remotely. If you *really* want to do this, set `NOREMOTE` to `false` in `/etc/athena/rc.conf` and `rm /etc/noremote`. Note that if `NOREMOTE` is set, people can still log in at the console; if you don't want them to be able to do so, set `NOCREATE`.

A third flag you may want to know about is `NOATTACH`, which is set and unset the same way as the other two flags. If it is set, home directories will not be `attach`ed.

5.2 Local Users

Although your machine will share Athena's user and group information, you can add local users and groups as well. To avoid conflicting with Athena, we recommend that you give local users and groups numerical IDs either less than 100 or greater than 32000, and names not currently used by Athena users or groups. Also, you'll have to add local user names to `/etc/localusers` if you want them to receive mail at your machine:

5.3 Local Mail Addresses

By default, mail sent to `blah@`*hostname*`.mit.edu` is treated as if sent to `blah@mit.edu`, where `blah` is any valid sequence of characters and *hostname* is your hostname. However, you can change this behavior by adding `blah` to `/etc/localusers`: `echo blah >> /etc/localusers`.

5.4 Local Home Directory

If you like, you can store your files on your local hard drive instead of on Athena. The advantage of this is that you do not have to rely on AFS servers working or be limited by your Athena quota. On the other hand, you will have to copy all of your files from your locker to your hard drive. You will then have to telnet to your machine anytime you want to access those files.

If you choose to keep a local home directory, note that many programs have configuration files stored in your home directory. For example, `discuss` creates a dotfile called "`.meetings`" in your home directory, and updates it each time you use `discuss`. These files will not be in sync with the ones in your Athena home directory. Keeping these in sync can be annoying. One technique is to symbolically link the files from your local home directory to your Athena home directory.

Another alternative is to export some local files via NFS. This approach makes the export more transparent, but has one major problem: NFS is **not** secure, and we recommend that you do not run an NFS server. If you really, *really* want to export files via NFS, however, there is a way to make NFS relatively secure: Make sure all lines in `/etc/exports` are of the form

```
/usr/export     *.mit.edu(root_squash,ro)
```

where `/usr/export` is the root of a tree you want to make publicly available.

The `*.mit.edu` restricts access to machines in the `mit.edu` domain, the `root_squash` remaps root to nobody and the `ro` makes the export read-only. Read the `exports` man page for more info. Be aware that using NFS at all can open up security holes; it is entirely conceivable that *anyone* could read *any* file on your system. (This may be a little paranoid, but you're much better off being too paranoid with NFS).

In general, it is probably easier (and safer!) just to use your Athena home directory.

5.5 Local Printers

If you're using RedHat and have installed all necessary packages (`lpr`, `printtool`, and probably `ghostscript`), you can run a program called `printtool`, which will take care of setting everything up for you. (Notes: `printtool` needs to be run as root and under X. `ghostscript` is not installed by default, so you will need to install it yourself if you do not have a PostScript printer.)

In order to be able to print to your printer from other workstations, you have to follow a more complicated procedure which includes getting a Hesiod entry for your printer. If you want to do this, please ask on `net-help@mit.edu` and somebody will fill you in on the details.

5.6 Running FTP and HTTP Servers

If you want, you can configure your machine as an FTP or HTTP server. Please remember that running these services may introduce security holes, and may also slow your machine down. If you still want to run them, though, it is easy to do so.

To run an FTP server, install the packages `wu-ftpd` and `anon-ftp`, and put the files you want to export in `/home/ftp/pub`. You also need to uncomment the line referring to ftpd in `/etc/inetd.conf` by removing the `#` at the beginning of the line and to restart `inetd` by running the command `kill -HUP` *pid* as root, where *pid* is the process ID of the running `inetd` process. (If you do not understand this, you can also just reboot after editing `/etc/inetd.conf`.

In general there is no reason to run a nonanonymous FTP server; you can always FTP to your account at `ftp.dialup.mit.edu` instead or use `rcp`. Nevertheless, it is possible; to do so, you have to follow the above instructions and then run `/usr/bin/passwd` to set a local password. (You need to do this because `ftpd` doesn't deal with Kerberos. For that same reason, you will not get tickets this way.)

To run an HTTP server, install the package `apache` and put the files you want to export in `/home/httpd/html`. (Icons can go in `/home/httpd/icons`, and CGI scripts[8] must go in `/home/httpd/cgi-bin`.)

6 Security

By now, you probably have a Linux machine which is on MITnet (and hence the Internet). One thing every person running a Unix machine on some network should note is that Unix and the network are probably both insecure. In general, Unix machines that are newly bought or newly set up are notoriously insecure. They are probably vulnerable to any number of attacks. `http://web.mit.edu/network/security.html` has general information about Unix system security at MIT. RedHat-Athena should be relatively safe, but there are still some things you should probably do, such as adding yourself to the `netusers` mailing list (`blanche netusers -a $USER`) and reading the `discuss` meeting `bloom-picayune:/usr/spool/discuss/linuxch-security`. Another good source for information about security holes in RedHat Linux is the WWW page `http://www.redhat.com/support/docs/rhl/rh42-errata-general.html`. (This page also contains information about other problems which RedHat Linux 4.2 may have.)

One other very important issue is configuring your machine to support secure telnets. After installation, your machine will accept incoming telnets; however, they will **not** be encrypted! This means that if you log in and type your password, you will be transmitting your password **in clear text** over the network! This means that anyone who is on the physical network(s) that your password is transmitted through will be able to obtain your password, if they are sufficiently motivated and knowledgeable.[9]

In order to support secure incoming telnets, you need what is called a `srvtab`. This file contains a secret key which allows your machine to authenticate incoming users to Kerberos and then encrypt the connection. In order to get this file, send mail to `accounts` asking for one; be sure to include both your username and your computer's hostname in the message. You should receive a response three to five days later telling you how to retrieve the file. You should install it in `/etc/athena/srvtab` and execute

```
chown root.root /etc/athena/srvtab
chmod 400 /etc/athena/srvtab
/usr/athena/bin/ksrvutil change
```

Your `srvtab` should start working in a day or two. If you require more assistance, send mail to `net-help`.

Remember that it is important to keep your system secure; if unauthorized users gain access to your machine, they may be able to damage your data and the data of other users of your system (including data stored on Athena accounts).

6.1 Changing Your Root Password

It is a good security measure to change your root password every so often. However, this procedure is currently difficult because of the way Athena deals with passwords and `passwd` files. Here's what you have to do:

- Log in as, or `su` to, root.
- Type `/usr/bin/passwd root` and enter a new root password.
- Edit `/etc/passwd.local` with your favorite text editor and replace the encrypted password following the string "`root:`" with the encrypted root password from `/etc/passwd`.[10] Important note: If you do not copy the encrypted password correctly, you will have trouble regaining root access on your system. It is best to use your editor's commands to copy it to avoid the possibility of error.

[8] CGI scripts may introduce security holes of their own. Be *very* careful with them.
[9] Look at `http://web.mit.edu/telnet/www/scenarios.html` for more information.
[10] An encrypted password consists of thirteen random-looking characters between the first and second colons on a line in a `passwd` file.

6.2 Kerberos Tickets

Kerberos tickets work on Linux the same way they work on other operating systems:

Normally, your tickets will be gotten for you during login (i.e., xlogin). If you ever want to get tickets manually, type `kinit` *username*, where *username* is your Athena username. Enter your Athena password when prompted. You can check to see that this was successful by running `klist`.

Your tickets will last for about ten hours and will become useless after this time. You can use the `renew` alias to get new tickets. (This command will work on any Athena workstation.)

If you manually get tickets, you should `kdestroy` them when you are done.

6.3 AFS Tokens

Like Kerberos tickets, AFS tokens contain authentication information and are automatically gotten for you during login and destroyed upon logout. However, they are not the same; as their name suggests, AFS tokens apply only to AFS, whereas Kerberos tickets are used for authentication to other MIT servers.

If you want to get AFS tokens manually, you can type `aklog`, optionally followed by a list of cells or paths to authenticate to. (For instance, `aklog sipb.mit.edu` will authenticate you to the SIPB cell, and `aklog /mit/sipb` will authenticate you to the SIPB locker.) You can verify that this worked by running the command `tokens`.

Your tokens should automatically expire at the same time your tickets do; if you need new tokens, `renew` will get them for you. (`renew` is the one command that works with both tickets and tokens.)

If you manually get tokens, you should run `unlog` when you are done to destroy them.

7 Backups

You should regularly back up any files you store on your local disk. Howver, there are some things you should know before making backups of your Linux system. First, do **not** attempt to back up remote file systems, particularly not `/afs`. You also should not back up `/proc`, as it is not a physical file system. There are two ways to do this; either back up each partition separately (the `-l` flag tells `tar` not to follow mount points; similar options may exist for other backup software) or explicitly exclude `/afs`, `/proc`, and other mount points.[11] Other things you probably don't want to backup are `/usr/vice/cache` (your AFS cache, whose contents are simply copies of files on AFS servers stored locally to save time) and directories named `tmp` (which stands for *temporary*).

Also, if there is any chance that somebody else can read your backup, and particularly if you are backing up over the network,[12] you should be sure to exclude `/etc/passwd*` and `/etc/athena/srvtab`. The contents of `/etc/passwd.local` tell malicious people who can telnet to your machine. (The contents also allow them to determine your root password, but only if you chose an easily-guessed password, which you shouldn't have done anyway, or if they have a lot of time or computational power.) If **anybody** gets a copy of your `srvtab`, however, the consequences can be dire. For instance, somebody with the contents of your `srvtab` could monitor encrypted connections to your machine, pretend to be your machine, and convince your machine that he has tickets as any user.

8 Feedback

You will encounter some problems. The programs we are providing may have bugs, the packages may be incomplete, and this documentation may be lacking. Please tell us what problems you encounter in the process of Athena-izing your Linux box so we may improve the process. Send your comments to `linux-dev@mit.edu`. If you have questions, send them to `linux-help@mit.edu`. If you find bugs in programs or the configuration of Linux-Athena, report them with `sendbug`.

This document is `$Revision: 1.26 $`. *Last modified* `$Date: 1998/08/25 02:17:24 $`

[11] You can exclude specific files with GNU `tar`: `tar cXf` *exclusionfile path dev* will back up all of the files under *path* which are not listed in *exclusionfile* to the device (or file) *dev*.

[12] Network backups can be eavesdropped on unless you encrypt the connection with a program like `ssh`. Make them only if you have no other choice.