TrinityOS: A Guide to Configuring Your Linux Server for Performance, Security, and Manageability

David A. Ranch `dranch at trinnet dot net`

May 22, 2005

TrinityOS and its associated archive scripts guide the Linux user in a step-by-step fashion using a common example throughout to configure over 50+ Internet services. The main focus of TrinityOS is to do this in a secure fashion while keeping both performance and manageability in mind. The documents also guide the user in other advanced topics such as aquiring their own Internet domain(s), moving DNS servers, confirming if you've been hacked, fighting SPAM email, and fixing various Linux file system, partition, LILO, and data recovery problems.

Contents

1 Copyright Notice

TrinityOS(TM)(c) `<http://www.ecst.csuchico.edu/dranch/LINUX/index.html##TrinityOS>`

Written, Maintained, Trademarked, and Copyrighted by David A. Ranch (dranch at trinnet dot net)

Sorry for all the legal stuff...

I've already had one company try to take the name TrinityOS from me (thus the trademark - Reg. Numbers 2440502 and 2525874). I also have had one LDP Guide author ("Securing and Optimizing Linux Red Hat Edition - A Hands on Guide") rip off a large portion of TrinityOS's content without even referencing me or TrinityOS as a source. Unfortunately, this author simply rewrote / rephrased the sections of it to avoid any direct copyright issue though the content is the same. So, with all this bad luck, I had to start covering my butt from the many lowlifes in the world.

Anyway, if you would like to use some of the content from TrinityOS in your project, you NEED to contact me first for permission. I'm an easy going guy so it won't be a big deal. Please just don't use my stuff first and ask second. That's pretty silly.

2 Introduction

TrinityOS is a complete Linux server configuration, maintenance, and security guide for the Linux novice and guru alike! Though there are a LOT of features covered in TrinityOS, you don't have to implement all of them. All I can say is, if you are going to connect your Linux box to the Internet, at least INSTALL the packet firewall!!

This document is tailored as a step-by-step, example driven document, instead of a detailed explanation doc on each Linux feature. It doesn't go into many debugging aspects since the Linux Documentation Project's (LDP) HOWTOs already cover this. The TrinityOS document is intended for a techincal audience but hopefully everything is laid out well enough that a new user should be able to follow along without too much trouble!

All of TrinityOS's step-by-step instructions, files, and scripts are fully scripted out for an automatic installation at:

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

* For the curious, the name TrinityOS and my company, Trinity Designs, is NOT derived from being religious (the holy Trinity). The name "Trinity Designs" came from the Trinity Alps in Northern California and "TrinityOS" came from the name of the first atomic bomb testing site in White Sands, New Mexico.

Like any UNIX document, it must be updated constantly to remain relevant. I will do my best to maintain this document but all comments, ideas, etc. are appreciated to keep TrinityOS valuable!

This guide was initially based off the Slackware v3.2 distribution but due to a disk crash, I then installed Redhat 5.0 to try it out. From that point on, I now try to make TrinityOS doc reflect other distributions.

Note: Most of the initial functionality given in this document is already available in a modern day distribution such as Mandrake, Redhat, Debian, SuSe, etc. If you are using any other distribution than Redhat, Debian, etc., you will need to use this doc as a *reference* or a project management guide only. You will then need to obtain the various software sources or binaries by hand and configure the software via its native methods.

** Please note that this document will always be "Under Construction". **

Everything in the "Current Features List" has been implemented and should be documented. Some things in the "Future Features" section have already been completed though not necessarily documented yet. If you have any specific questions about the "Future" or "Current features".. feel free to ask!

Tangent #### # # If you have come to this doc directly, you also might want to # check out the rest of my WWW page at: # # `<http://www.ecst.csuchico.edu/dranch>` # # It covers other topics such as:

- Who am I (Why did I do all this?)

- Linux (TrinityOS, book reviews, other links, etc)
- PC Hardware (PC chipsets, CDR evals, BIOS discussions, etc)
- RAS technologies (xDSL, 56K modems, PPP optimizations, etc)
- Cable modems (how they work, the system I setup, @Home, etc)
- ISDN technologies (T/A & router evaluations, etc)
- Researching ISPs (How to pick a good ISP)
- Bookmarks (Check out my extensive WWW bookmarks)

```
**********************************************************************
** Would you like to be notified when I update my WWW page or       **
**   specifically the TrinityOS doc?                                 **
**                                                                  **
** Every "update" e-mail is based from both the ChangeLog WWW page  **
** and the TrinityOS ChangeLog section so you will know what        **
** exactly was updated without any extra fluff.                     **
**                                                                  **
** If you're interested, send an e-mail to                          **
**                                                                  **
**                  mailto:dranch at trinnet dot net               **
**                                                                  **
** with a subject of "Add me to your updates list" and I'll add     **
** you to the list!                                                 **
**                                                                  **
** -P.S.- In the same request email, tell me what specifically you  **
**       were/are looking for on my WWW page or in TrinityOS.       **
**       I'm always taking new requests for additions and expanded  **
**       coverage of topics already on my page.                     **
**                                                                  **
**       So don't be shy!                                           **
**********************************************************************
```

3 Feature Sets

3.1 Current Features:

3.1.1 Master References and Recommended Guidelines

- An extensive URL library and current version list for all installed and recommended Linux tools and applications
- Example guidelines on documenting the hardware and partition layout of your specific hardware

3.1.2 Linux Distribution Thoughts:

- Thoughts and recommendations on picking a Linux distribution
- A common "Search & Replace" example template throughout the document for both better clarity and the ability to use Search/Replace tools to customize this doc to YOUR specific setup

3.1.3 Core OS setup:

- Configuring, compiling, installing, and booting both a 2.2.x & 2.0.x kernel
- Lilo configuration, security, and recovery
- PCMCIA / CARDBUS PC-Card Services
- Software RAID 0 (striping) hard drives
- 7-CD SCSI CD-ROM changer system
- Automated Patching via RPM notifiers
- EXT2 file system tuning
- IDE hard drive performance optimization
- Dual printing system support for both UNIX and Windows/Samba hosts

3.1.4 Network Connectivity:

- Strong, configurable, and well commented IPCHAINS and IPFWADM packet firewall rule sets for SINGLE, DUAL, and THREE NIC environments. This section also incluides a complete intro on how Packet and Stateful Inspected firewalls work
- Automated rollback script for the loading of rc.firewall rule sets so that if you make an error in the firewall rule set and the rule set doesn't complete execution, a backup rule set will be automatically loaded to restore connectivity.
- Full LAN masquerading (NAT or Network Address Translation) using private IP addressing
- Masq IP port forwarding support (PORTFW)
- Three Ethernet network card support setup and TCP/IP Performance optimization (modem and cable modem users w/ DMZ support)
- DNS servers running both primary and secondary zones using Bind in a CHROOTed and and SPLIT Zone configuration
- Full Sendmail-based SMTP and backup SMTP e-mail system support w/ domain masquerading & Anti-SPAM measures with support for more than one Internet domain on one EMAIL server
- IMAP4 / POP3 remote email service
- DHCPd server for other LAN machines (laptops, etc)
- DHCPc Linux client setup for getting TCP/IP addresses
- SAMBA: Full Microsoft Windows file & printing support
- NFS: Full Sun RPC-based Network File System support
- IPSEC (Swan) VPN [Almost Complete]
- PPTP VPN client and forwarding through IPMASQ
- HTTPd WWW server support
- PPP connectivity for primary PPP connectivity AND backup PPP connections

- Dial-on-Demand (Diald) Internet connections (modem users) - Automatic Internet connections every 15 minutes (modem users)
- Direct dial-in terminal / PPP access via a modem
- NTP time calibration
- Full UNIX printing via LPR

3.1.5 Security:

- Complete physical and OS-level security recommendations and guidelines
- Full SSHd (encrypted TELNET) support
- Actively Updated Linux system security and patching (Shadow passwords, etc)
- Advanced SYSLOG logging and nightly filtered reports emailed to the root user
- Prioritized TrinityOS "CRITICALITY" rating system in the CHANGELOG section to gauge the level of urgency of security vulnerabilities, system mis-configurations, etc.
- NMAP port scanning to test your packet firewall
- Anonymized Sendmail Banners

3.1.6 System backup:

- Minimum backups to floppy
- Full backups via Hard drives or to tape using BRU with emergency restore diskette creation
- Full APC SmartUPS power down support (APCUPSd) with both paging support and plotting power stats with GNU Plot to a graph which is emailed via "Sendlogs"
- Backing up the server to a CD-R [not completed yet]

3.1.7 More extensive guides:

- How to fix LILO, HD partitioning, and file system corruption
- How to obtain an Internet domain(s) via a domain registrar
- How to successfully move Internet domains across DNS servers and/or TCP/IP addresses
- How to recover from your box being hacked and how to RE-secure it
- Full documentation on how understand and FIGHT all that SPAM email
- How to understand and fight SPAM email
- SSH encrypted PORTFW VPN tunnels for email, etc

3.2 Future Features:

(Won't be implemented in any particular order)

3.2.1 * TrinityOS TO-DOs:

- Add more "Configuration via GUI tools" sections

3.2.2 * Network stuff

- Give instructions on compiling Xntp
- Modularize the rc.firewall rulset so updates can be transparent and not require additional tailoring for each update.
- Remove LPR and replace it with LPRng or CUPS
- IPv6: Configure and setup IPv6 and possibly setup a IPv6 tunnel via the 6Bone
- Dial Backup: Add automatic analog modem dial backup when the ADSL/Cable modem goes down
- CODA: Replace NFS support with CODA
- Add a CACHING only setup for DNS
- Setup a email list server (MajorDomo, Petidomo, dunno yet)
- Email sent dynamic IP address exception requests for access through the TCP Wrappers and the IPFWADM rule sets
- DHCPc client setup for Cablemodems
- 128-bit encrypted Apache SSL WWW server
- Move over to xinetd for better DoS protection
- WWW Proxy services
- WWW banner add filtering
- Give instructions on compiling Xntp

3.2.3 * Security Stuff

- Replace the Sendlogs script to use either Swatch or LogSentry
- Automate the firewall hits logging for trend analysis
- Install PGP / GPG for secure and/or verified communications to: other users, Internic, binaries/source code verification, etc.
- Tripwire Security Breech monitoring [not completed yet]
- SATAN / SAINT / Nessus / COPS / ISS security testing

3.2.4 * Application stuff

- Get Sendmail to run in an SMRSH shell
- Implement Procmail to do local email filtering
- Setup fetchmail to get remote email vs. setting up a remote .forward

3.2.5 * Administration stuff

- Rotate the UPS logs
- Implement automatic weekly incremental tape backups to a tape drive.

3.2.6 * System Stuff

- Iomega parallel ZIP drive support

4 Hardware Configuration

This document uses methodologies that I have developed over the years. Some of these docs have saved my butt on several occasions (documenting things like Drive partition maps, I/O and IRQ maps). This may seem like a pain in the butt to do initially but when you need them..

YOU NEED THEM!

4.1 - Distribution:

- Mandrake 7.0 w/ all available patches

4.2 - Kernel

v2.2.25

4.3 Hardware Used:

```
  - Intel Pentium 200Mhz / 128MB EDO RAM

  - Intel TC430HX motherboard (cannot tune IRQ use)
       - Serial port #1: COM1 - IRQ 4
       - Serial port #2: COM2 - IRQ 3
       - LPT1                 - IRQ 7
       - IDE 0                (disabled)
       - IDE 1                - IRQ 15

  - Network:
       Eth0: Compaq Netelligent 10/100 Dual port (PCI) - port #1 (IRQ  11)
             - cable modem side

       Eth1: Compaq Netelligent 10/100 Dual port (PCI) - port #2 (IRQ  14)
             - Int LAN

  - Video:
       Matrox Millennium II (4MB) - (PCI)

  - Sound:
       Built-in Windows Sound System (IO:530h, IRQ: 9, L-DMA: 0, H-DMA: 1,
```

```
               MPU: 330h, MPU IRQ: -1


    - Controllers:
         - Adaptec 2940UW SCSI controller (PCI) - IRQ: 10
               - Used for SCSI disks (ext. cabling to RAID enclosure)

         - Adaptec 2940U SCSI controller (PCI)  - IRQ: 14
               - Used for CDROMs and Tape drives (int. & ext. cabling)

    - I/O Adapter - (ISA)
         (2) port serial / (1) parallel
         - COM3 - IRQ 4
         - COM4 - IRQ 3
         - LPT2 - IRQ 5


    - Storage Devices:
                                == In the primary system case ==

                - HDC:   Maxtor DiamondMax+    10.0GB (UDMA)[512k][LBA] [
                - HDD:   IBM 120GB HD

                - SR0-6: Nakamichi 7-CD 2x changer (ID: 4)
                - SR7:   Philips CM4xx 4x CDROM    (ID: 5)
                - ST0:   HP T4000 TR4 Tape drive   (ID: 6) [dead?]

                == In the secondary RAID enclosure ==

                - SDA:  Seagate ST39173N 9GB (20Mb/s) (ID: 0) - Primary HD
                - SDB:  Seagate ST39173N 9GB (20Mb/s) (ID: 1) -
                - SDC:  IBM DNES-309170  9GB (20Mb/s) (ID: 2) -
                - SDD:  Seagate ST39173N 9GB (20Mb/s) (ID: 3) - dd backup of SDA


                        - I/O:(See docs on IRQTUNE to better understand why these
                                 are like this.  It makes a difference!)

                                ttyS0: COM1 - APC SmartUPS UPS
                                ttyS1: COM2 - N/A
                                ttyS3: COM3 - USR Courier v.Everything
                                ttyS2: COM4 -

                                LPT1:  Hp LaserJet-IIp  (UNIX & Samba share)
                                LPT2:  Canon S800       (UNIX & Samba share)



------ I/O Maps and "Expert" fdisk partition tables -----
```

```
IRQ Map:

         0: timer                 (system)
         1: keyboard          (system)
         2: Cascade               (system)
         3: COM2-N/A          (Motheboard) & COM4-
         4: COM1-APC Smartups (Motherboard & COM3-US Robotics modem
         5: Sound             (Motherboard)
         6: Floppy            (system)
         7: LPT1-printer      (motherboard)
         8: Clock             (system)
         9: Cascade
        10: Adaptec 2940U     (PCI)
        11: Compaq Ethernet#1 (PCI)
        12: PS/2 mouse        (motherboard)
        13: Math coprocessor
        14: Adaptec 2940UW    (PCI)
        15: IDE1              (motherboard)

I/O Port MAP:

        170-1F7h:       IDE1
        1F0-1F7h:       IDE0
        200-207h:       (not used) usually Joystick
        278-27Fh:       LPT1
        2E8-2EFh:       COM4
        2F8-2FFh:       COM2
        330-331h:       Windows Sound Systye Pro MPU-401
        376-376h:       IDE1
        378-37Fh:       LPT1
        3E8-3EFh:       COM3
        3F0-3F5h:       Floppy drive
        3F6-3F6h:       IDE0
        530-533h:   Windows Sound System

        E800h:  AHA2940U
        EC80h:  AHA2940U
        FCE0:   TLAN #1
        FCF0:   TLAN #2
        E400h:  System BIOS
        E800h:  Systen BIOS
        F000h:  System BIOS

DMA Map:

        0 - Windows Sound System
        1 - Windows Sound System
        2 - Alternative Floppy DMA
        3 - Floppy DMA
        4 - Casecade
```

```
        5 - None
        6 - None


-----
All hard Drive partition tables
-----


/dev/hdc (normal mode printout - expert truncates)
==================================================
Disk /dev/hdc: 16 heads, 63 sectors, 19390 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hdc1            1        1    19390  9772528+  83  Linux native
==================================================


/dev/sda (expert mode printout)
==================================================
Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders

Nr AF  Hd Sec  Cyl  Hd Sec  Cyl   Start    Size ID
 1 80   1   1    0 254  63    6      63  112392 06
 2 00   0   1    7 254  63 1023  11245517655435 05
 3 00   0   0    0   0   0    0       0       0 00
 4 00   0   0    0   0   0    0       0       0 00
 5 00   1   1    7 254  63  261      63 4096512 83
 6 00   1   1  262 254  63  294      63  530082 82
 7 00   1   1  295 254  63 1023      6312289662 83
 8 00 254  63 1023 254  63 1023      63  738927 83
==================================================


/dev/sdb (expert mode printout)
==================================================
Disk /dev/sdb: 255 heads, 63 sectors, 1106 cylinders

Nr AF  Hd Sec  Cyl  Hd Sec  Cyl   Start    Size ID
 1 00   1   1    0 254  63 1023      6317767827 83
 2 00   0   0    0   0   0    0       0       0 00
 3 00   0   0    0   0   0    0       0       0 00
 4 00   0   0    0   0   0    0       0       0 00
==================================================


/dev/sdc (expert mode printout)
==================================================
Disk /dev/sdc: 255 heads, 63 sectors, 1115 cylinders
```

```
Nr AF  Hd Sec  Cyl  Hd Sec  Cyl   Start    Size ID
 1 00   1   1    0 254  63 1023      6317912412 83
 2 00   0   0    0   0   0    0       0       0 00
 3 00   0   0    0   0   0    0       0       0 00
 4 00   0   0    0   0   0    0       0       0 00
==================================================


/dev/sdd (expert mode printout)
==================================================
Disk /dev/sdd: 255 heads, 63 sectors, 1106 cylinders

Nr AF  Hd Sec  Cyl  Hd Sec  Cyl   Start    Size ID
 1 80   1   1    0 254  63    6      63  112392 06
 2 00   0   1    7 254  63 1023  11245517655435 05
 3 00   0   0    0   0   0    0       0       0 00
 4 00   0   0    0   0   0    0       0       0 00
 5 00   1   1    7 254  63  261      63 4096512 83
 6 00   1   1  262 254  63  294      63  530082 82
 7 00   1   1  295 254  63 1023      6312289662 83
 8 00 254  63 1023 254  63 1023      63  738927 83
==================================================

-------

--
```

5 Software URL download map and checklist

- Software recommended and used for the TrinityOS doc (roughly in this order).

  ** NOTE** Put all code in /usr/src/archive/

  I personally recommend to putting ALL additional software source code, RPMs, etc in /usr/src/archive. In the "archive" directory, I make subdirectories for the various code like dns, ssh, sendmail, etc. This IS your box though so put things ANYWHERE you so wish. :)

5.1 Master site for all Internet RFCs:

- `<http://www.cis.ohio-state.edu/rfc/>`

5.2 The Master IANA site

- For all Internet port numbers, protocol numbers, etc. A VERY recommended place to go, download them ALL, and put them in /etc/iana.

  - `<http://www.iana.org/numbers.htm>`

  To create a local copy, do the following:

```
mkdir /etc/iana
cd /etc/iana/
wget -r -l 1 -nH --no-parent http://www.iana.org/numbers.htm
```

5.3 Master site for all known Internet Trojan ports

- `<http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html>`

5.4 Distribution Sites and Update MIRRORS:

Any Service Packs, security patches, etc. for your installed Slackware or Redhat distribution(s)

5.4.1 Mandrake Updates:

- Master URL: `<http://www.linux-mandrake.com/en/security/>`

5.4.2 Redhat Updates:

- Master MIRROR URL: `<http://www.redhat.com/mirrors.html>`
- Fast: `<ftp://ftp.codemeta.com/pub/mirrors/redhat/updates/>`;
- 5.2 only: `<ftp://ftp.infomagic.com/pub/mirrors/linux/RedHatUpdates/>`

5.5 Newest stable kernel

`<ftp://ftp.kernel.org/pub/linux/kernel/>` or `<ftp://ftp.freesoftware.com/pub/linux/sunsite/kernel/>`

5.5.1 2.6.x

- 2.6.11.10 is stable

5.5.2 2.4.x

- 2.4.30 is stable
  - All kernels less that 2.4.20 have the lcall7 local DoS attack vunerability. No REMOTE DoS attack is possible.
  - All kernels less than 2.4.13 have a serious symlink vunerability. Please upgrade your kernel.
  - Please note that the 2.4.x series of kernels is still quite new and some aspects of it are immature in comparison to 2.2.x kernels (PCMCIA, Power Management, etc). But, several new aspects of the 2.4.x kernels might make you want to try it (faster IP stack, stateful firewalls, journaled filesystems, etc.)

5.5.3 2.2.x

- 2.2.26 is stable
  - All versions less than 2.2.22 have a local denial of service risk though no REMOTE DoS attack is possible.
  - ALL versions less than 2.2.16 have a TCP exploit that when combined with tools such as Sendmail, will leed to a root compromise.
  - All kernels below 2.2.12 have a IP fragmentation bug. This will make ALL strong IPCHAINS rule sets vulnerable!
  - 2.2.11 has a memory leak issue.

5.5.4 2.0.x

- 2.0.40 is stable
  - Any lower version have a DoS attack against the TCP/IP stack

5.6 IP NAT, MASQ, Load Balancing, and High Availability tools

- There are several implementations but here are the common ones:
  - A Good Master Reference to the various NAT implimentations for multiple Operating Systems
    - `<http://www.uq.net.au/~zzdmacka/the-nat-page/>`
  - Main Linux NAT, Load Balancing, and High Availability reference site:
    - `<http://www.linas.org/linux/load.html>`
  - Newer NAT implementations:
    - IPROUTE2: The primary true Many:Many NAT implimentation for 2.2.x kernels - `<ftp://ftp.inr.ac.ru/>`
      - Mirror: `<ftp://ftp.tux.org/people/alexey-kuznetsov/ip-routing/>`
      - Documentation #1: `<ftp://post.tepkom.ru/pub/vol2/Linux/docs/>`
      - Documentation #2: `<http://www.compendium.com.ar/policy-routing.txt>`
      - Advanced Routing HOWTO: This doc covers IPROUTE2, Policy-based routing (source IP), GRE tunnels, Multicast, Queueing, etc, and more - `<http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html>`
    - An older NAT implimentation available here: `<http://proxy.iinchina.net/wensong/ipnat/>`
  - Excellent tutorials on Linux NAT and the home of one of the first implementations:
    - `<http://www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat.html>` or
    - `<http://www.suse.de/mha/HyperNews/get/linux-ip-nat.html>`

5.6.1 MASQ E-mail list : By far the BEST way to get MASQ-help (very helpful!!)

- Send mail to `<mailto:masq-request@tiffany.indyramp.com>`

5.6.2 Linux IP Masq

2.4.x kernels

- NetFilter now provides for both 1:Many Masq-like NAT and true 1:1 NAT:
  - `<http://www.netfilter.org/documentation/index.html>`

2.2.x kernels

- NOTE: ALL versions less than 2.2.16 have a IP fragmentation bug (among other things). This will make ALL strong IPCHAINS rule sets vulnerable! Upgrade NOW!
  - IPCHAINS Main site:
    - `<http://www.netfilter.org/ipchains/>`

  IPMASQADM port forward patches:
  - `<http://ipmasq.webhop.net/juanjox/>` or
  - `<ftp://ftp.compsoc.net/users/steve/ipportfw/linux21/>`

  The beginnings of Stateful Inspection for Linux:
  - 2.0.x kernels
    - `<http://www.ifi.unizh.ch/ikm/SINUS/firewall.html>`
  - 2.1.x / 2.2.x kernels
    - `<ftp://ftp.interlinx.bc.ca/pub/spf>`

2.0.x kernels

- IPFWADM (source must download regardless if installed with Redhat)
  - Slackware:
    - `<ftp://ftp.xos.nl/pub/linux/ipfwadm/ipfwadm-2.3.0.tar.gz>`
  - Redhat:
    - `<ftp://ftp.xos.nl/pub/linux/ipfwadm/ipfwadm-2.3.0-1.src.rpm>`
- IPFWADM patches (if required for pre-2.0.30 kernels) at:
  - `<http://ipmasq.cjb.net/ipfwadm-2.3.0-generic-timeout.patch.gz>`
- IPCHAINS support for the 2.0.3x kernels
  - `<http://aemiaif.lip6.fr/willy/pub/linux-patches/ipnat/>`
  - `<http://www-miaif.lip6.fr/willy/pub/linux-patches/>`
- IPPORTFW Port forwarding for 2.0.x kernels
  - Homepage:
    - `<http://www.ox.compsoc.org.uk/steve/portforwarding.html>`
  - Patches:
    - `<ftp://ftp.ox.compsoc.org.uk/pub/users/steve/ipsubs/sub-patch-1.37.gz>`

- Interpreting Firewall hits:
  - This is a great URL in addition to the content in Section 10 on how to interpret your firewall logs and what all the information means:
    - `<http://www.robertgraham.com/pubs/firewall-seen.html>`

5.7 PPP - v2.4.3 (not needed for most cable modem users)

Primary site: `<http://www.samba.org/ppp/index.html/>`

5.8 ML/PPP

- PPPd now supports ML/PPP as of 2.4.x (see above)
- Strong Implimentation: `<http://mp.mansol.net.au/mp/>`
- Lots of data, little code: `<ftp://ftp.east.telecom.kz/pub/src/networking/ppp/multilink>`
- Another implementation (runs on 2.2.x+ and he is looking for testers) `<http://linux-mp.terz.de>`
- Dead link? `<http://mp.ins-coin.de>`

5.9 PPPoE (PPP over Ethernet) : Needed for some DSL and Cablemodem users

Very popular user-space client : Primary Site: `<http://www.roaringpenguin.com/pppoe.html>`

Kernel-Space client known for somewhat better performance: `<http://www.davin.ottawa.on.ca/pppoe/>`

Some other informational URLs as well:

`<http://www.suse.de/~bk/PPPoE-project.html>`

`<http://www.sympaticousers.org/faq.htm>`

5.10 Diald v1.00 (not needed for cable modem users)

Diald is now maintained by a new author and site:

`<http://diald.sourceforge.net>`

RPMS: `<http://ipmasq.webhop.net/juanjox/>`

Download the original Diald and Diald patches (Diald v0.16.5)

`<http://www.loonie.net/eschenk/diald.html>`

5.11 Bind / Named current: 9.3.1 and 8.4.6

Sources: `<ftp://ftp.isc.org/isc/bind/src/>`

Versions: 9.2.2 requires non-vulnverable OpenSSL code. It's also recommend to download both the source code /and/ the associated .asc PGP signature for that version of BIND.

RPMs: Finding new RPMs for the newest versions of Bind isn't very easy. Once place you might have luck is the CONTRIB area of sites like Redhat and Mandrake. Those RPMs seem to work fine but some people do NOT trust someone else's compiled code, so, it's your choice.

`<ftp://rawhide.redhat.com/>`

You can also find a chroot-ed version of bind here:

`<ftp://ftp.fi.muni.cz/pub/users/kas/bind-chroot/>`

Announcement list:

Send email to bind-announce-request@isc.org with "subscribe" in the subject field.

5.12 Vlock (stock in Redhat if installed)

`<ftp://ftp.freesoftware.com/pub/linux/sunsite/utils/console/vlock-1.0.tar.gz>`

5.13 Network Sniffers

5.13.1 - TCPDUMP (stock in Redhat if installed) - Excellent network packet sniffer

`<ftp://ftp.freesoftware.com/pub/linux/sunsite/system/network/management/>` or `<ftp://ftp.ee.lbl.gov/tcpdump.tar.Z>`

5.13.2 - IPtraf - Excellent high level network protocol watcher

- Current 2.7.0

`<http://iptraf.seul.org>`

5.13.3 - EtherReal - An excellent GUI decoder

- Current 0.10.11

`<http://ethereal.zing.org/>`

5.14 Sendmail current: v8.13.4, v8.12.11, and v8.11.7

`<ftp://ftp.sendmail.org/pub/sendmail/>`

Both Sendmail 8.12.9 and 8.11.7 are secure though they have a problem with the "smrsh" shell. TrinityOS doesn't use this but if you are concerned about it, a patch is available. Currently, if you plan to use 8.11.x, you need to run 8.11.7 secure it from a few recently found remote root exploits.

RPMs: The newest Sendmail is NOT available in RPM form from sendmail.org but it IS in Redhat's CONTRIB area. It seems to work fine but some people do NOT trust someone else's compiled code, so, it's your choice.

`<ftp://ftp.infomagic.com/pub/mirrors/linux/RedHatContrib/libc6/i386>`

Announcement list:

Send an email to majordomo@Lists.Sendmail.ORG with the text "subscribe sendmail-announce" in the body of the message.

5.15 POPAuth

I have taken over ownership of these documents but haven't had a chance to post them yet. If you would like to get a copy of them, please email *me* `<mailto:dranchattrinnetatnet>`

For allowing remote POP-3 clients to be able to use the SMTP server to send email.

5.16 Virtual Email domains

To support multple email domains w/ Sendmail, Qmail, etc check out:

`<http://www.linuxdoc.org/HOWTO/Virtual-Services-HOWTO.html>`

5.17 DHCP Server - DHCPd v3.0.2

DHCP Faq: `<http://www.dhcp-handbook.com/dhcp_faq.html##hddhs>`

RFC Info: `<http://www.dhcp.org/rfc2131.html>`

`<http://www.dhcp.org/rfc2132.html>`

Legacy Info: `<http://www.cis.ohio-state.edu/rfc/rfc1542.txt>`

Download: `<http://www.isc.org/dhcp.html>`

5.18 DHCP Client

DHCP HOWTO: `<http://www.tldp.org/HOWTO/mini/DHCP/index.html>`

dhclient v3.0.2 comes with the server code above

DHCPcd 1.3.22-p14: `<http://www.phystech.com/download/dhcpcd.html>`

Other DHCP info:

`<http://www.linux-firewall-tools.com/linux/firewall/index.html>`

A HOWTO specific to the RoadRunner Cablemodem setup, but it's still a good site: `<http://www.vortech.net/rrlinux/>`

5.19 WU-FTP v2.6.2 - with multiple patches

FTP: `<ftp://ftp.wu-ftpd.org/pub/wu-ftpd/>`

FAQ: `<http://www.cetis.hvu.nl/koos/wu-ftpd-faq.html>`

5.20 NetWatch

`<ftp://ftp.digital.com/pub/linux/redhat/powertools-5.0/i386/>`

5.21 Getdate (NTP) - v1.2 (Was SETTIME)

`<ftp://metalab.unc.edu/pub/Linux/system/network/misc/getdate_rfc868-1.2.tar.gz>`

5.22 NTP Clock Sources

`<http://www.eecis.udel.edu/mills/ntp>`

5.23 Tape Back up:

- BRU (it's not free but it's the best Linux backup software out there IMHO. This is one place you just CAN'T skimp!) Recommended!

http://www.estinc.com

5.24 Mozilla v1.7.8 (Netscape is dead)

Original Mozilla (deprecated) - 1.7.8 Firefox - 1.0.4 Thunderbird - 1.0.2

`<ftp://ftp.mozilla.org>`

5.25 SSH

Commonly used BSD licensed OpenSSH client/server (totally free) - current: 4.0p1 `<http://www.openssh.com/>`

Original Commercial SSH.com client/server (free for Linux :: for now) - current: 3.2.6.1 `<http://ftp.ssh.com/pub/ssh/>`

Additional UNIX SSH tunneling URLs:

`<http://www.ccs.neu.edu/groups/systems/howto/howto-sshtunnel.html>`

5.26 MDADM and Raidtools

MDADM v1.11.0): `<http://www.cse.unsw.edu.au/~neilb/source/mdadm/>`

Good but old info on Linux RAID: `<http://linas.org/linux/raid.html>`

Raidtools (DEPRECATED) 1.00.3: `<http://people.redhat.com/mingo/raidtools/>`

5.27 Samba current: 3.0.14a (stock in most distros if installed)

`<http://www.samba.org>`

Also, they have great docs at `<http://samba.anu.edu.au/>`

5.28 PCMCIA Services - 3.2.8

`<http://pcmcia-cs.sourceforge.net/>`

5.29 UPS software - APCUPSd and Powerchute

Original and quite nice APCUPSd open-source daemon - v3.10.17a: `<http://www.apcupsd.com/>` or `<http://www.sibbald.com/apcupsd/>`

Official APC Powerchute for Linux - v4.5.3 - Free closed-source daemon with excellent Xwindows support: `<http://www.apcc.com/tools/download/index.cfm>`

5.30 Apache WWW server - 2.0.54 and 1.3.33

Standard Apache: `<http://www.apache.org>` or `<ftp://ftp.redhat.com/pub/contrib/i386/apache-1.2.6-5.i386.rpm>`

SSL-encrypted Apache:

`<http://www.apache-ssl.com/>`

5.31 File Integrity testing/Monitoring

5.31.1 TripWire:

Tripwire has gone OpenSource for LINUX! Woohoo! Though it isn't available quite yet, it will be there soon:

`<http://www.tripwire.org>`

Also, as of v2.2.1, Tripwire now runs on Glibc.

`<http://www.tripwiresecurity.com/products/Tripwire_ASR20.cfml>`

You can also get the older versions here:

`<ftp://coast.cs.purdue.edu/pub/COAST/Tripwire>`

5.31.2 Aide:

AIDE is a GNU version of Tripwire - v0.10

`<http://sourceforge.net/projects/aide>`

5.31.3 ViperDB:

ViperDB is another GNU version of Tripwire

`<http://www.resentment.org/projects/viperdb/index.html>`

5.32 RPM update tools:

5.32.1 AutoRPM current version: 1.9.8.1

`<http://www.kaybee.org/kirk/html/linux.html>`

5.32.2 The Perl module "Libbet"

`<http://cpan.valueclick.com/modules/by-module/Net/>`

5.32.3 RPM Watch current version: 1.1

(does not work for Redhat 5.2+) [Will be phased out] `<ftp://ftp.iaehv.nl/pub/users/grimaldo/rpmwatch-1.1-1.noarch.rpm>`

5.32.4 RPMLevel (from the author of RPMWatch)

`<http://coralys.com/products/>`

5.33 Mkisofs

`<ftp://ftp.fokus.gmd.de/pub/unix/cdrecord/mkisofs/>`

5.34 Compression tools

BZip2 : `<http://sourceware.cygnus.com/bzip2/index.html>`

5.35 Bash HOWTO

`<http://www.linuxdoc.org/HOWTO/Bash-Prompt-HOWTO.html>` Also see 42 (Section 42) in TrinityOS

5.36 Dial-In Server HOWTO

`<http://www.swcp.com/jgentry>`

5.37 SWAN / IPSEC VPN

Project home page:

`<http://www.xs4all.nl/freeswan>` or `<http://www.flora.org/freeswan/>`

SWAN email list:

`<http://www.xs4all.nl/freeswan>`

Overview `<http://www.cygnus.com/gnu/swan.html>`

Download the IPSec code from:

Broken? `<ftp://ftp.xs4all.nl/pub/crypto/freeswan>`

Works ? `<http://ftp.xs4all.nl/pub/crypto/freeswan>`

or

`<http://www.flora.org/freeswan/download>`

Other Mini-HOWTOs:

https://www.seifried.org/articles/ipsec/

5.38 PPTP VPNs and client software

- Client: `<http://sourceforge.net/projects/pptpclient/pptp-linux-1.1.0-1.tar.gz>`
- PPP shim: `<http://sourceforge.net/projects/pptpclient/ppp-mppe-2.4.0-4.tar.gz>`
- Additional docs: `<http://pptpclient.sourceforge.net/howto.html##setup>`
- Addition troubleshooting: `<http://pptpclient.sourceforge.net/howto-diagnosis.phtml>`
- IPMASQ patches: `<ftp://ftp.rubyriver.com/pub/jhardin/masquerade/ip_masq_vpn.html>`

5.39 PGP Email Encryption

- PGP: `<http://web.mit.edu/network/pgp.html>`

5.40 Serial consoles and Remote TELNET

- Remote Serial HOWTO (for more details on configuring serial consoles): `<http://tldp.org/HOWTO/Remote-Serial-Console-HOWTO/>`

5.41 IP logger

`<ftp://ftp.tu-graz.ac.at/pub/linux/redhat-contrib/SRPMS/iplogger-0.1-1.src.rpm>`

5.42 Hardware Performance Tuning:

- PowerTweak - optimize the BIOS/Chipset/PCI registers `<http://powertweak.sourceforge.net/>`
- Preempt patch - make the kernel more responsive under load `<http://www.tech9.net/rml/linux/>`
- IRQTune - optimize IRQ response times - good for PPP/Modem users `<ftp://shell5.ba.best.com/pub/cae/irqtune.tgz>`
- HDparm - good for hardcore IDE performance users `<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/diskdrives>`

5.43 Security Documentation, Tools, and Resources

5.43.1 Various Security Mailing lists and documentation

- `<http://www.shmoo.com>`

5.43.2 The Linux Security HOWTO

- `<http://www.linuxdoc.org/HOWTO/Security-HOWTO.html>`

5.43.3 Logging tools:

- CheckLogs:
  - `<http://www.iae.nl/users/grimaldo/chklogs.shtml>`
- Swatch:
  - `<ftp://ftp.stanford.edu/general/security-tools/swatch>`
- Psionic LogCheck:
  - `<http://www.psionic.com/abacus/logcheck>`
- LogSurfer: (like Swatch but with state checking!)
  - `<http://www.cert.dfn.de/eng/logsurf/home.html>`

5.43.4 - Nmap - v3.81 :

`<http://www.insecure.org/nmap/>`

5.43.5 - Nessus - 2.24 :

`<http://www.nessus.org/>`

5.43.6 - COPS (old)

`<ftp://ftp.freesoftware.com/pub/linux/sunsite/system/security/cops_104.tgz>`

5.43.7 - Saint (new version of Satan)

`<http://www.wwdsi.com/saint/>`

5.43.8 - SATAN (Old)

Newer: `<ftp://ftp.porcupine.org/pub/security/index.html>`

Older `<ftp://ftp.win.tue.nl/pub/security/satan.tar.Z>`

5.43.9 - Solar buffer-overflow fixer

`<ftp://ftp.huwig.de/pub/linux/mama/2.0/stack_noexec-symlink-security-fix.bz2>`

5.43.10 - Kurt Seifried's Linux Administrators Security Guide (LASG)

`<https://www.seifried.org/lasg/>`

5.43.11 - Ofir Arkin's paper on ICMP protocol fingerprinting

`<http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.0.pdf>`

5.43.12 - Other URLs:

Test Exploits: `<http://www-miaif.lip6.fr/willy/security/>`

Test Exploits: `<http://www.rootshell.org>`

Test Exploits: `<http://www.l0pht.com>`

Test Exploits: `<http://www.geek-girl.com>`

Security Alerts: Subscribe to BugTraq at `<mailto://LISTSERV@NETSPACE.ORG>`

More Security:

`<http://www.ecst.csuchico.edu/dranch/LINUX/index-linux.html##security>`

`<http://www.ecst.csuchico.edu/jtmurphy/>`

5.43.13 - Abacus Security Initiative

Includes host_sentry, port_sentry and logchecker.

`<http://www.psionic.com/abacus>`

5.43.14 - Intrusion Detection Systems (IDS) Tools SHADOW (SANS)

SHADOW (SANS): `<http://www.nswc.navy.mil/ISSEC/CID/step.htm>`

Snort: `<http://www.snort.com>`

5.43.15 - Network Flight Recorder

Setup HOWTO: `<http://www.nswc.navy.mil/ISSEC/CID/nfr.htm>`

NFR software: `<http://www.nfr.net/download/>`

NFR ID Attack ID Packages: `<http://www.nswc.navy.mil/ISSEC/CID/nfr_id.tar.gz>` `<http://www.l0pht.com/NFR/>`

5.44 WWW proxy (Apache or Squid)

5.45 WWW Ad banner filtering

`<http://www-math.uni-paderborn.de/axel/NoShit/index.html>`

patch: `<http://www.america.com/chrisf/web/NoShit/WebFilter_0.5.patch.gz>`

Example filter: `<http://www.america.com/chrisf/web/NoShit/library.txt>`

5.46 Zip drive

`<http://www.torque.net/campbell>`

5.47 Linux Applications:

`<http://www.xnet.com/blatura/linapps.shtml>`

5.48 Linux Games:

X-Shipwars: `<http://fox.mit.edu/xsw/>`

5.49 Linux Instant Messenger clients:

- GAIM 1.3.0 `<http://gaim.sf.net>`
- Reviews of different IMs for Linux: `<http://www.linuxnetmag.com/en/issue2/m2icq1.html>` `<http://www.portup.com/gyandl/>`

6 Thoughts on Picking a Linux Distribution

6.1 - Installing Linux distribution

This is too complicated to be completely covered in TrinityOS. But, to get you started, here are a few comments that talk about what Linux distribution might be right for you.

One thing I've been asked over and over is regarding users that are trying out Linux with an old Linux CD (given to them, etc.). With the new 2.4.x kernels out, all the newest Linux distributions BLOW AWAY the old ones in terms of ease of setup, performance, hardware compatibility, etc. So, I recommend that you get a new copy a given Linux distribution and give that a look. And you can't tell me it's expensive when you can get almost ANY Linux distribution for under $3.00 US a CD from places like `<http://www.cheapbytes.com>`.

```
*----------------------------------------------------------------------------*
*  What do I use?  I currently use Mandrake v9.1 on my work laptop (Dell) and *
*                                                                            *
*  7.0 at home but I'm worried about Mandrake's direction (see more below)    *
*----------------------------------------------------------------------------*
```

So, with that behind us, here is a few notes:

6.2 Redhat: http://www.redhat.com

Redhat has recently discontinued both their regular Linux distribution via retail channels as well as their downloadable ISO version (currently 9.0). Moving forward, Redhat has created two projects. The "Fedora" project which is an opensource distribution and then their Redhat Enterprise Linux v3.0 distro line. A good question is if the Fedora project will take over where the RH9.0 distro left off in terms of quality, etc. I have no idea but I do know that the testing won't be nearly as good and I doubt the installer and GUI tools will be as refined as they've been in the past.

Fedora: The main differentiation with with the two RH distros is there isn't any Redhat commercial grade testing or tech support for the Fedora version This is no different than using distros like Debian, Gentoo, etc. which are well supported by the Linux community as a whole. All Fedora support will be via web forums, 3rd party support vendors, etc.

Enterprise Linux: The RH Enterprise Linux line offers email/phone support for 2-3 years for email/phone support and 5 years for critical security patches, etc. which is very good in my option. Unfortunately, the Enterprise line comes in three versions (workstation only (WS), small server (ES), and big server (AS)) and thus charges accordingly:

As of November, 2003 ——————— WS - $180 - only initial install support :: Full 1 yr support is $299 US. - NO servers support - this is only a workstation (very limiting)

ES - $350 - only initial install support :: Full 1 yr support is $799 US.A - Full servers support - Dual SMP only - limited RPM package list

AS - $1500 - support included but 4 CPU version starts at $2500 US. - Full servers support - 4way CPU + - more complete RPM package list

Yes, this is expensive for a enduser but not bad for an enterprise setup. BUT, my major gripe with RHEL is that the software package list or RPM list Linux is probably < 50% that of RH 9.0 was. Check it out, here is a full list of the RHEL ES 3.0 RPMs - `<http://www.ecst.csuchico.edu/~dranch/LINUX/Rhel/>` As you can tell, not only does this make EL expensive but you don't get a whole lot for your money other than a good software patch policy.

Anyway, Redhat has been a premier Linux distribution that has a strong installation tool and has some great system administration utilities too. One of the best parts of Redhat is its increamental RPM package installation and upgrade system. Redhat is constantly upgraded, they even support / offer patches for their oldest distro versions, and it is well supported in the Linux community.

Redhat is a good choice for the Linux newbie that wants a more server-focused distro or a GUI configuration approach running with all kinds functionality. Don't let the server focus fool you.. this distro is very desktop friendly as well. Redhat is a Gnome shop vs. a KDE-centric distro.

If you are already a UNIX snob, you might find Redhat's layout a little wierd (unless you are a Sun Solaris (SYSV) person - the /etc/rc.d/rc2.d layout is similar).

BUT, many people don't like Redhat. Why?

1. Redhat has a LOT of extra software built-in. Yes, you can choose the "Custom" installation process and get rid of most of the options (recommended) but a FULL install is quite large (a full RH8.0 install is 4.6GB!). Yes, you can pick a "custom" install and reduce the number of installed packages but it's still a heavy distro.

2. If you want to *learn* UNIX (not specifically Linux) in the classic LINUX step-by-step fashion and truly understand it (the hardest but BEST way (IMHO)), Redhat probably wouldn't be my first choice! Yet, I do have to admit my opinion is slowly changing though.

3. Redhat changes the entire behavior of how Linux is set up and configured compared to other distributions like Slackware to be more easy to use, modifible via scripts, etc. Unfortunately, Redhat's GUI tools don't easily tell you what it is going to do to your config files. If you want to learn UNIX in a classic fashion, go with Slackware or, to a lesser extent, Debian, SuSe, etc! Those distributions are a LOT more plain and easier to initially figure out.

4. RPM Hell. You've might have heard about this term before. What this basically means is that if you want install a given program, sometimes it has prerequisite of installing another program first. Ok, so you try to install that required program to only find thhat this sub-required program might have THREE other required programs. Then when you try to install the sub-sub programs, they TOO have requirements. Get the idea? Though it is always solved with patience (using RPM manually and installing all the required programs), many people hate RPMs for this reason. Fortunately, Redhat's newest RPM GUI tools determine all the required other programs for youi. Some say this is a fundamental flaw of the RPM system itself. I don't think it's that bad but I'm a patient kind of guy (most of the time at least).

All Newer versions of Redhat have enhanced installation programs for simple installations but with the ability to configure advanced options like software RAID, LVM, etc. Also, the ASCII, NCURSES, and X-Windows versions of the "linuxconf" and "control-panel" GUI interfaces are getting VERY cool!

6.3 Mandrake: http://www.linux-mandrake.com

Mandrake Linux, currently at version 9.2, is a close derivative of Redhat Linux with some significant changes and add-ons. The main difference between Mandrake and Redhat (even today) is that Mandrake is compiled for [Pentium] or newer machines. Redhat is currently compiled for Intel 386 (i386) processors though their kernels are optimized. With the Pentium optimizations alone, Mandrake can yeild anywhere from a 10-20% performance increase over RedHat on some platforms.

Next, Mandrake has been adding more customized tools to their distribution. With these tools, like the "Mandrake Updater", administration is easier. If you like GUI tools, Mandrake has them!

One thing I do want to mention is that Mandrake installers within the "Drak" have become very powerful. The installers are very simple for the newbie but can also be very powerful (installtion of software RAID, LVM, etc). Mandrake is also very security conscious and gives the user the option of different default security settings, etc.

Much like Redhat, Mandrake also shares with the RPM hell problem. Fortunately, Mandrake has RPMdrake which determines all of the required dependancies for you and fixes most of this issue.

One last thing that must be noted is that like most Linux vendors, Mandrake has changed their patch support policies. They now only offers security patches for ONE year from the release of the distro. After that, you MUST upgrade to their newest distro. The alternative is to buy their Corporate Server version which is pretty expensive (Corp. Server 1.1 is $799) but will give you support 2+ years. In comparison to Redhat and SuSe's support policies, Mandrake is both expensive and lacking equal support. This pains me as I'm a big Mandrake fan but servers need to be supported and upgrading every two years is silly. Ultimately, if it's a server that you don't plan on upgrading very often, getting the Corporate version might make sense. For a destop system, only getting patches for 1 year sucks but then again, newer distros will have more featuress, etc.

6.4 SuSE: http://www.suse.com

SuSE, currently in version 9.0, is a powerful distribution from Germany. I had previously tried their older releases but there was so much embedded German text in it, it bothered me so I gave up on it. I recently installed newer versions and it seemed much better. The installation program is pretty good though I think Redhat or Mandrake's is better. But, SuSE has a nice configuration tool called YaST and they were one of the first to come with the KDE window manager.

If you like the BSD style of configuring services (much like Slackware, FreeBSD, etc.), you'll like SuSe.

BUT.. recently, Novell with a grant from IBM is trying to buy SuSe. What will this mean to SuSe? Good question but it will take them a while to improve or bury it.

6.5 Debian: http://www.debian.org

Debian is currently on their 3.0R1 release and though I haven't used Debian much, many people out there (mostly power users) seem to like it a lot. Debian is a community distro which means that there is no "Debian" corporation trying to make money at it. It's run and maintained by the community so the distro is only as good as the contributors. It has been best described to me as as a distribution that old Slackware users will LOVE which hate Redhat. Interestingly enough, the defunct Corel and Storm distributions were based on Debian.

Debian doesn't include the kitchen sink in for software like Mandrake or Redhat but it's laid out in a good manner and it has it's own RPM-like installation/upgrade system called dPKG with GUI frontends like "apt" or the older too, "dselect". One thing to note about Debian's package system is that unlike the "RPM hell" situation (see the Redhat section above), it can automatically determine a package's dependancies (what other programs are needed to get this particular program to run) and automatically download AND install the required packages. In this respect, Debian is still untouched in ease of use.

Like Redhat, Debian is reported to be constantly updated and well supported. Many people argue that Debian is even better updated than Redhat though they are considerably slower to release new distributions with the newest versions of Gnome, KDE, etc. compared to the other distro vendors.

6.6 Gentoo: http://www.gentoo.org/

Gentoo is a new distro community distro that is very similar to Debian in the respect that there is no "Gentoo" corporation trying to make money from it. It's run and maintained by the community so the distro is only as good as the contributors.

Fortunately, Gentoo brings something new to the Linux distro mix. Most traditional linux distros (Redhat, Mandrake, SuSe, etc.) all install pre-compiled binaries which makes the installation quick and painless but the resulting distro might not take advantage of your hardware (ahem.. Redhat). Gentoo takes a totally different stance on the installation phase. Specifically, after you pick the packages you want to install, Gentoo will compile ALL of them from the sources to maximize your hardware. This is great though a full installation can take DAYS if not even a WEEK or more depending on how fast your hardware is and how many packages you are installing.

Once installed, Gentoo uses the "portage" program installation system which is similar to the *BSD "ports" system. This is where everything is compiled from source. It's a pretty easy system to use as it automatically figures out where to download the programs from and how to compile them. It just is time consuming. But, the sweetest aspect to "portage" system is that with one command, you can upgrade your ENTIRE distro install to the current versions of all packages with ONE command! Very powerful though I also consider this dangerous too (config files change, too many variables if something breaks, etc.)

6.7 Slackware: http://www.slackware.com

Slackware, now at version 9.1 is one of the original Linux distributions and it is still one of my favorites. It definately isn't as slick in terms of installation or functionality compared to Mandrake but it's laid out in a clear manner. The INIT scripts (the scripts that are executed to bring the system up) are laid out in a very readable fashion (BSD-style - So is SuSe) and everything is obvious (in the open). Slackware will be a comfortable fit for the UNIX guru peoples out there.

Like Redhat, Slackware uses a software package system (pkg) for modularized system upgrades. Though it isn't as fancy as Redhat's RPM system.. it has almost all the same functionality. Though patches do come out for Slackware, Redhat's community usually has patches available FASTER.

6.8 Caldera: http://www.calderasystems.com/

Caldera or SCO, now at v3.1, is the most commercial of all the Linux distributions. They initially pulled ahead of the pack with a better installation program and auto-installing hardware modules but almost everyone has caught up pretty quickly. Caldera was understood to have one of the easiest installation program of ALL the distributions though Mandrake might have them beat now.

Caldera differentiates itself by trying to meet the needs of the corporate market. For example, they have completed a port of Novell's NDS directory services to Linux. Pretty cool!

But, it should be noted that SCO seems to be taking on Linux on the legal front. They are sueing various companies for Millions if not Billions of dollars. In my opinion, this is a last gasp for them to stay alive but this isn't a way to keep the Linux community happy with them.

6.9 Other Distributions

There are other Distributions out there to pick from depending on your hardware platform (Dec Alpha, Motorola PowerPC, etc) such as:

TurboLinux - popular in Japan / Network clusters

LinuxPPc `<http://www.linuxppc.org>` - for PowerPC machines

LinuxPro `<http://www.wgs.com/>`

LinuxWare `<http://www.trans-am.com/>`

MkLinux `<http://www.mklinux.apple.com/>` - For 680x0 and PPC Apples

Stampede `<http://www.stampede.org/>`

You'll have to experiment and ask other Linux people what distribution they like and WHY! Personally, I'd recommend to get one of those multiple Distrobution CD sets from places like `<http://www.cheapbytes.com>` and try them out yourself!!

For more Distribution details, check out:

`<http://www.linux.org/dist/english.html>`

`<http://www.tldp.org/HOWTO/CD-Distributions-EN-HOWTO/index.html>`

`<http://www.linuxgazette.com/issue31/hughes.html>`

7 Installing a distribution, patching it, and doing a Search/Replace on TrinityOS

7.1 Upgrading/Updating your Linux distribution:

Like ANY Linux distribution, bug fixes, security releases, etc. are always coming out and you NEED to stay on top of it. Remember, Linux is very functional but without a given security patch, a hacker can break into your box and do ANYTHING! Redhat, Debian, Slackware, etc have their own incremental update systems that makes this easier.

P.S. If the program you update to with "pkgadd" has different configuration file layouts, you will have to the conversion manually. Debian and Redhat's systems can do the conversion for you though I've had mixed results with this.

7.1.1 Redhat users:

Go to the Redhat Updates URL in 5 (Section 5) and download all the recent patches to a directory (ie. /tmp/patches). Once you have all of the newest RPMs, you should use the "Fresh" option of the RPM tool. This will update the RPMs on your machine ONLY if an older version of the RPM is installed on your machine. So, I recommend thast you do:

rpm -Fvh /tmp/patches/*

Also, please heed these following warnings regarding RPMs:

```
*******************************************************************************
** Don't always trust RPMs!!!!                                               **
**                                                                           **
**  See [Section 50] for more specific instructions on how to use  **
**  RPMs, see what files will be installed/replaced/OVERWRITTEN BEFORE you   **
**  install them, etc.                                                       **
*******************************************************************************
** Staying on top of new RP Ms                                               **
**                                                                           **
**  You should also implement the RPM notification tool that is documented   **
**  in [Section 43] to stay on-top of this in the future!          **
*******************************************************************************
```

7.2 TrinityOS diagrams and Search and Replace Keys

——————————————————-

This is how the TrinityOS network is laid out:

–

Network topology diagram:

```
 --------
/        \
|Internet >------------------+
\________/                   |
                         Cablemodem
                             |
                   +-----------------------+
                   |         |             |
                   |  External Link: eth0  |
                   |  IP:  100.200.0.212   |
 ---------         |  DGW: 100.200.0.1     |
/ Various \        |                       |
|  Remote |        |     ------------      |
|  Sites   >-ISDN--|- External Link: ppp0  |
|    &    |        |  IP: dynamic          |
| Internet|        |     ------------      |
|   link  |        |     DMZ Link: eth2 ---|----< To 802.11b wireless network
\  backup /        |    IP: 192.168.10.1   |            IP: 192.168.10.x
 ---------         |     ------------      |           DGW: 192.168.10.1
                   |                       |           DNS: 192.168.10.1
                   |  Internal Link: eth1  |
                   |  IP: 192.168.0.1      |
                   |          |            |
                   +-----------------------+
                              |
                      8-port 100Mb/s switch
                              |
          +----+----+----+----+----+----+----+----+
          |    |    |    |    |    |    |    |    |
         PC   PC   PC   PC   PC   PC   PC   PC   PC
         #1   #2   #3   #4   #5   #6   #7   #8   #9
          |
          |
       /----------------\
        IP: 192.168.0.2
         DGW: 192.168.0.1
         DNS: 192.168.0.1
```

- Next, this section is to custom tailor your copy of TrinityOS to your specific environment. Do a search/replace on the "Search for" fields and replace them with your correct "replace with" fields.

PLEASE NOTE: If you are going to use IP Masquerading, you should use one of the private address spaces as described in RFC 1918 `<http://www.cis.ohio-state.edu/htbin/rfc/rfc1918.html>` such as:

- Class-A: 10.x.x.x
- Class-B: 172.16-31.x.x
- Class-C: 192.168.x.x

	search for	replace with (given as an example)
Your main login ID	johndoe	your-login
Your PPP ISP name	your-ppp-isp-name	your-ppp-isp-name
Your PPP ISP #	555-1212	555-1234
Your PPP login	your-ppp-login	your-ppp-login
Your PPP password	your-ppp-passwd	your-ppp-passwd
The Linux machine name	roadrunner	your-linux-boxes-name
Domain Name	acme123.com	yourdomain.org
Second Domain Name	another-domain.com	yourseconddomain.org
Internal IP network	192.168.0.0	192.168.0.0
Internal IP address	192.168.0.10	192.168.0.10
Internal gateway IP	192.168.0.1	192.168.0.1
Internal broadcast IP	192.168.0.255	192.168.0.255
Internal DMZ IP network	192.168.10.0	192.168.10.0
Internal DMZ IP address	192.168.10.10	192.168.10.10
Internal DMZ gateway IP	192.168.10.1	192.168.10.1
Internal broadcast DMZ IP	192.168.10.255	192.168.10.255
External IP network	100.200.0.0	100.201.0.0
External IP address	100.200.0.212	100.201.0.212
External gateway IP	100.200.0.1	100.201.0.1
External broadcast IP	100.200.0.255	100.201.0.255
Remote SECONDARY DNS	ns.backupacme.com	ns.yourdomain.org
External secondary DNS	102.200.0.25	102.201.0.25
Reverse DNS lookup	54.44.80.10	50.0.201.102
Explict allowed IP#1	200.211.0.40	200.244.0.40
Explict allowed IP#2	200.211.0.41	200.244.0.41
Explict allowed IP#3	200.211.0.42	200.244.0.42
Explict allowed IP#4	200.211.0.43	200.244.0.43
ISP DNS server #1	10.200.200.69	10.222.222.44
ISP DNS server #2	10.200.200.96	10.222.222.88
Your SMB Workgroup:	ACME123	your-linux-boxes-SMB-workgroup-name
Your pager email:	1234567@skytel.com	2321432342@skytel.com
An internal PORTFWed MASQ machine name:	coyote	one-internal-MASQed-machine-name

```
A internal PORTFWed
MASQ machine IP:          192.168.0.20           192.168.0.20

Internal machines
  allowed to connect
  to the MASQ server:     192.168.0.11           192.168.0.11
                          192.168.0.12           192.168.0.12

Remote PPTP setup
  PPTP server running at: MyEmployer.com         MyEmployer.com
  PPTP server IP:         220.1.2.3              220.1.2.3
  PPTP username:          YourUserNameHERE       YourUserNameHERE
  PPTP CHAP name:         REMOTE-PPTP-CHAP-HERE  REMOTE-PPTP-CHAP-HERE
```

7.3 ## Fixing Redhat, Mandrake, etc. (bugs) that are right out of the BOX! (ouch!):

* These are errors, bugs, annoyances, etc that I've notice in Redhat5.x. But, these might be fixed in later CD releases, patches, etc.

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

7.3.1 - Fix all cron permissions (some fixed in RH6.x)

```
chmod -R 750 /etc/cron.hourly
chmod -R 750 /etc/cron.hourly/*
chmod -R 750 /etc/cron.daily
chmod -R 750 /etc/cron.daily/*
chmod -R 750 /etc/cron.weekly
chmod -R 750 /etc/cron.weekly/*
chmod -R 750 /etc/cron.monthly
chmod -R 750 /etc/cron.monthly/*
```

7.3.2 - Let Minicom and "ls" run in Color:

- Edit /etc/profile and add:
  - Add the following after the "export" line if you have Minicom installed: MINICOM="-c on" export MINICOM
  - This "ls" issue is fixed in RH6.x but its good to setup regardless. Edit the /etc/bashrc file and add:

    alias ls='ls –color=yes'

7.3.3 - Let ColorGCC always run to make compiling a little more obvious

- Add the following to the */etc/bashrc* file to make compiling highlight various warnings, errors, etc. I think it helps..

```
        export CC="colorgcc"
```

7.3.4 Fix the timezone

- NOTE: This is supposed to be already fixed in a Glibc RPM fix
  - Edit the /etc/profile file
    - Just above the "EXPORT PATH" line, add the line for Pacific Daylight time (adjust for your Time zone) TZ=PST8PDT
      Now edit the "EXPORT PATH" line and append the word "TZ"

7.3.5 - Change the default UMASK (default file/directory create)

NOTE: Changing this behavior makes the permissions of all NEWLY created files only readable by certain users and groups. This can have a detrimental effect on programs that need to be used by multiple users. The default is "umask 002 else umask 022".

NOTE2: If you see two "umask" lines, change them BOTH to 027

- edit /etc/profile, find the umask line(s) and make them it read "umask 027"

7.3.6 - Fix compressed FTP downloads (still broken in RH6.1)

NOTE: The changes were:

- "compress" is in /usr/bin and NOT /bin
- I had previously patched TAR to understand .BZ2 compression but this is now already done in RH6.x and most other modern Linux distributions (the man pages don't reflect this. Obviously this is STILL a bug as of Mandrake 7.0.).
- If you have an old distribution, compile up the new tar executale. Then put this new TAR binary in /usr/local/bin.
- Create a link to the new tar file ln -s /usr/local/bin/tar /bin/tar
- Now, to fix FTP so you can get compressed archives automatically from ftpd, edit the /etc/ftpconversions file and make it look like this:

```
:.Z: :  :/usr/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
:   : :.Z:/usr/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
:.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
:   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
:   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
:   : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
```

7.3.7 - Fix the permissions on the /etc/rc.d/init.d script files!!!

Bad, Bad, Bad. Only "root" and admin groups should be able to do this type of adminstration.

```
                        chmod -R 770 /etc/rc.d/init.d/*


================================================================================
```

---

8 Initial System security

This covers CMOS setups, disable ports, TCP wrappers, shadow passwds, etc.

First thing, I would recommend to do in addition to following TrinityOS for your needed purposes, read LDP's Security HOWTO for a more detailed explanation of what to do. Interestingly enough, I never read it until recently and a LOT of things I had independantly recommend was already in the Security HOWTO too! So, it sounds like we are on-track! I recommend you read it too! The URL is in 5 (Section 5).

8.1 BIOS/CMOS Settings

Upon system boot, enter into the CMOS setup

- AMI BIOSes use the DEL key
- Compaq BIOSes use the F10 key
- Some Pheonix BIOSes use Control-Escape, Control-Alt-Ret, F2, or Control-Alt-Shift (mostly in vendor-customized versions such as Dell).
- IBM Series 300 uses F2 in their SurePath Bios.

- Once you are in the BIOS, search around and try to set the following:

8.1.1 + Enabled the BIOS password

- I recommend the combination of upper and lower case characters with numbers!

8.1.2 + DISABLE booting from the floppy drive

By changing the BIOS boot order from A:,C: to C:,A:

If you are extra paranoid, you can set the floppy drive to READ only or even disable the floppy drive all together if you wish.

8.2 Linux root Password

- Now, boot back into Linux and make sure you have a password for the root login

---

```
        passwd root
```

NOTE: You may not have noticed this but most Linux distributions only took the first -8- characters of your password. After that, they simply ignore ALL other passwords. For example, these two passwords are the SAME to Linux:

Pl3a5eGet0ut and Pl3a5eGe

Because of this, you need a strong password and it can ONLY be 8-characters long. You REALLY should use a combination of UPPER and lower case characters, numbers, and special characters such as:

[‘~!@#$%^&*()-_=+{[]}\|’";:,<.>/?]

Fortunately enough, the newer Linux distributions have fixed this issue. But regardless if this has been fixed on your distribution or not, it IS important that you choose a strong passwd.

8.3 Enable the "sticky" bit in /tmp

This ensures that only the file's owner can delete

a given file in /tmp (Fixed in RH6.x):

```
                chmod 1777 /tmp
```

8.4 - Disable the Control-Alt-Delete keyboard shutdown command

- This is pretty important if you don't have the best physical security on the box:

- Do implement this, edit /etc/inittab and change the line:

```
                ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

to

```
                #ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

- Now, for the system to understand the change, type in the following at a prompt

```
                /sbin/init q
```

8.5 - Disable the ability to run INIT in interactive mode

Newer Redhat:

- Edit the /etc/sysconfig/init script and change the line:

```
       prompt=yes
```

to..

```
       prompt=no
```

8.6 - Compile / install vlock (available in most modern distributions).

NOTE: Use this command if you are logged in as root and want to LOCK the ttys without having to log fully out and back in again. Nice!

8.7 - Change what system daemons get loaded by editing the following files in "/etc/rc.d/"

NOTE: Regardless of Linux distribution, you might want to SKIP some of the following steps if you plan to run:

- Samba (smb)
- Printing (lpd)
- Mail (Sendmail),
- NFS
- etc.

8.7.1 Redhat:

(though this is specific to Redhat, the following is a good read for ALL Linux users.)

The way that Redhat boots is the SysV way. This is where the OS will execute ALL files for a given runlevel (see definition below) that start with a "S" (that's a CAPITAL "S") and have a number after that in a numerical order from lowest to highest. For example, it will run "S10network" before it runs "S30syslog".

So what's a RUN-level? A run-level is the mode that the machine will load various system programs. Though this varies from Unix to Unix (Linux, Solaris, AIX, HP-UX, etc.), they are similar. For Linux, this is the run-levels (from /etc/inittab):

Please note that some Linux distributions have slight variations:

- 0: halt (stops the OS and sometimes shuts the power off)
- 1: single user (doesn't bring up the network, no passwd for root. Needed for system problems, lost root passwds, etc)
- 2: Redhat: Multiuser (Brings up the whole OS but doesn't mount remote file systems (NFS, CODA, etc)

  SuSe: Full Multiuser (Brings up the whole OS with any remote file systems)
- 3: Redhat: Full Multiuser (Brings up the whole OS with any remote file systems)

  SuSe: Xwindows (Brings up the system immediately into X-windows)
- 4: Unused
- 5: X-windows (Brings up the system immediately into X-windows)
- 6: Reboot (reboots the machine; usually into a COLD boot state [counts all the RAM, etc])

Also, if you didn't already notice, all of the files in various runlevel directories like /etc/rc.d/rc0, 1, 2, 3, 4, 5, 6.d are actually just symbolic links to all the real script files in /etc/rc.d/init.d! This makes things more manageable.

So, since Linux usually runs in multi-user / non-Xwindows mode, that means runlevel "3" will execute all files in the /etc/rc.d/rc3.d directory. Then, the system will begin to run ALL files starting with "S" in order. When you shutdown or restart the machine, you change the machine into runlevel "0" or "1". This will first execute all commands from the initial runlevel directory of "3" starting with "K". If the given process isn't already running, like my example for LPD, it will just skip it and move on. Get it?

8.7.2 Slackware:

The way that Slackware boots is the BSD way. It will execute the /etc/rc.d/rc.inet1 (network interfaces) file first. Then, it will run the /etc/rc.d/rc.inet2 (network services) file. This is much more readable than the Redhat method but its harder to maintain (IMHO).

8.7.3 Securing your machine by limiting what daemons load:

BSD-Style: Edit the following files in /etc/rc.d/ and make these changes unless you need that service.

```
                - rc.M (disable email and WWW servers)

                        - line 75:      #'d out all lines for Sendmail
                        - line 97:      #'d out all lines for httpd

                - rc.inet2 (disable SERVER and NFS servers)
                        - line 14:      #'d out all lines for lpd
                        - line 15:      #'d out all lines for lpd
                        - line 31:      #'d out all lines for portmap
                        - line 72:      #'d out all lines for mountd, nfsd, pcnfsd, bwnfsd
```

There are at least (6) ways to turn on/off what daemons load:

Via A GUI interface:

This process manipulation can be done either via:

- "chkconfig" command line utility
- "ntsysv" Ncurses GUI utility
- "tksysv" Xwindows GUI utility
- "control-panel" or "linuxconf" Xwindows GUIs.
- "Manual editing"
- "Deleting the package altogether"

Note - Though I'm a command line bigot, I feel the "ntsysv" GUI is the fastest way to modify these options!

NOTE #2 - It should be noted that some people really feel that if you are going to disable a package, you might as well REMOVE IT. This is technically MORE secure (nothing to run an exploit against) nor does it take up any disk space. Personally, I usually side with functionality and rather just disable the service vs. delete it all together. Now, if you're sure that you'll NEVER use this service, definately recommend to delete the package.

To DELETE a given package:

To remove packages:

- Redhat: rpm -e package-name
- Slackware: pkgdel package-name

NOTE #3 - I've found that when you first run these GUI tools, they will default to running and disabling some processes they SHOULDN'T! So, be careful and make sure that the tool is starting/stopping the correct daemons. Confirm this by going into the correct runlevel directory, say /etc/rc.d/rc3.d, and making sure only the minimal S* files are there.

With "chkconfig":

Please note that there might be some daemons that are missing and/or extra in your specific /etc/rc.d/init.d directory so make sure you enable/disable the appropriate ones for your needs.

```
--
#Disable automounters
chkconfig --level 2345 amd off

#Disable unless this is a laptop
chkconfig --level 2345 apmd off

#Disable unless you want to run batch programs within certain loads
chkconfig --level 2345 atd off

#Disable unless you want emails of EVERY ARP on your network segment
chkconfig --level 2345 arpwatch off

#Disable unless you want boot diskless workstations
chkconfig --level 2345 bootparamd off

#Disable unless this machine will be a DHCP *SERVER*
chkconfig --level 2345 dhcpd off

#Disable unless this machine will be a full blown router
chkconfig --level 2345 gated off

#Disable unless this machine will be a WWW server
chkconfig --level 2345 httpd off

#Disable unless this machine uses a modularized kernel
#  NOTE:  Not needed for 2.2.x+ kernels
chkconfig --level 2345 kerneld off

#Disable unless you really want to configure remote machines via Linuxconf
chkconfig --level 2345 linuxconf off

#Disable unless this machine will be a print server
#(for the local or remote machine)
chkconfig --level 2345 lpd off

#Disable unless you really need the proprietary MC server
chkconfig --level 2345 mcserv off

#Disable unless this machine will be a database server
chkconfig --level 2345 mysql off
```

```
#Disable unless this machine will be a caching or full blown DNS server
chkconfig --level 2345 named off

#Disable unless this machine will be a NFS server
chkconfig --level 2345 nfs off

#Disable unless this machine is a laptop or the PC has PCMCIA cards
chkconfig --level 2345 pcmcia off

#Disable unless this machine will be an NFS server or needs RPC tools
chkconfig --level 2345 portmap off

#Disable all R-cmds
chkconfig --level 2345 rusersd off
chkconfig --level 2345 rwalld off
chkconfig --level 2345 rwhod off

#Disable unless this machine is a email server
chkconfig --level 345 sendmail off

#Disable unless this machine is a Samba (MS File&Print) server
chkconfig --level 345 smb off

#Disable unless this machine is to support SNMP
chkconfig --level 2345 snmpd off

#Disable unless this machine is a local/remote HTTP proxy server
chkconfig --level 2345 squid off

#Disable unless this machine will be running X-windows
chkconfig --level 2345 xfs off

#Disable unless this machine will be an NTP server
chkconfig --level 2345 xntpd off

#Disable unless this machine will be part of a NIS/YP domain
chkconfig --level 2345 ypbind off
chkconfig --level 2345 yppasswdd off

#Disable unless this machine will be a NIS/YP server
chkconfig --level 2345 ypserv off
```

Manually:

NOTE: only do this to the processes you WON'T use.

NOTE #2: If, for some reason, any of the K or S* files don't exist and you want them to be there, use one of the GUI tools above.

Do this in /etc/rc.d/rc2.d, /etc/rc.d/rc3.d, and /etc/rc.d/rc5.d

---

```
- mv S08autofs K08autofs
```

```
                        - mv S20nfs K20nfs
                                        (unless this is for a full or caching NFS server)
                        - mv S20rusersd K20rusersd
                        - mv S20rwalld K20rwalld
                        - mv S20rwhod K20rwhod
                        - mv S30mcserv K30mcserv
                        - mv S98kerneld K98kerneld
                        - mv S35smb K35smb              (unless this is for a Samba F&P server)
                        - mv S60lpd K60lpd              (unless this is for a print server)
                        - mv S65portmap K65portmap      (unless this is for a NFS server)
                        - mv S95nfsfs K95nfsfs          (unless this is for a NFS server)
                        - mv S45pcmcia K45pcmcia        (unless this for a laptop)
                        - mv S65dhcpd K65dhcpd          (unless this is for a DHCP server)
                        - mv S85httpd K85httpd          (unless this is for a WWW server)
                        - mv S80sendmail K80sendmail    (unless this is for a mail server)
```

8.8 Shutting down most of inetd / xinetd

Inetd and Xinetd are called the "super servers" as they load a network server based upon a request from the network. I personally recommend that any service that you DON'T need shouldn't be able to load. This both minimizes CPU and Memory load as well as greatly reduces your security risk.

```
* The exceptions that I leave in and secure via a firewall and
* TCPwrappers are:
*
*       TELNET, FTP, SSH, sometimes TALK, POP-3, IMAP, and maybe FINGER.
*
```

Newer Linux distributions no longer use "inetd" but instead use a newer version called "xinetd". This new version allows for much more granular configuration as well as superior logging, etc. Overall, I really recommend Xinetd though it does take a little time to get used to.

XINETD: ——- Go into the /etc/xinetd.d directory and edit each of the files in that directoru. In each one of the service files that should be disabled, make sure that a line reading "disable = yes" is present. For example

/etc/xinetd.d/chargen

```
# default: off
# description: A chargen server. This is the tcp \
# version.

service chargen
{
    type        = INTERNAL
    id      = chargen-stream
    socket_type = stream
    protocol    = tcp
    user        = root
    wait        = no
    disable     = yes
}
```

I recommend to disable the following services and any other services enabled in your machine that you don’t need (unless noted below).

- chargen
- chargen-udp
- daytime
- daytime-udp
- echo
- echo
- finger (you might want to enable this)
- imap (you might want to enable this)
- ident (don’t enable this unless you use IRC)
- ipop3 (you might want to enable this)
- ntalk (you might want to enable this)
- swat
- talk (you might want to enable this)
- time

To make the change take effect, type in:

- Redhat: /etc/rc.d/inet.d/xinetd restart
- Slackware: kill -HUP ‘ps aux | grep xinetd | grep -v -e grep | awk ’{print $2}’‘

INETD: —— I recommend to edit the /etc/inetd.conf file and place a "#" in front of the lines to disable them (if not already done).

- echo - basic network functions that AREN’T needed
- discard - "
- chargen - "
- daytime - For checking the date remotely (or)
- time - "
- shell - Remote Shell. flexible but VERY insecure. A part of the R-command tools
- login - "
- exec - "
- comsat - Email box monitoring server (very old)
- talk - UNIX Talk (I usually allow this but secure it via the firewall/tcp-wrappers

- ntalk - "
- dtalk - "
- pop-2 - For checking email. Use POP3 instead.
- uucp - For sending/receiving email the OLD way.
- tftp - For simple file transfers (unless you need this functionality)
- bootps - For simple configuration transfer (very old; replaced by DHCP)
- cfingerd - For probing information on a specific user or who is logged in
- systat - For probing information about the system itself
- netstat - For probing information about the system's network
- auth - For the ident system to see what user is creating specific network traffic
-
- linuxconf - For remotely configuring the system via the Linuxconf GUI
- swat - For remotely configuring the Samba server via Swat

As noted above for Xinetd, some items you might want to leave enabled. Some you might want to leave available until you install a secure alternative like SSH):

- ftp - For insecure file transfer
- telnet - For insecure remote logins
- talk - For accepting local/remote real-time talk sessions
- ntalk - "
- dtalk - "
- pop-3 - For downloading email.
- imap - For checking email on the server.
- finger - For checking out info on system users (most people should disable this)
- cfinger - "
- NOTE: If you need to run finger, change the word "root" to "nobody".

Once you make these changes, finish editing the file. To make the change take effect, type in:

- Redhat: killall -HUP inetd
- Slackware: kill -HUP ‘ps aux | grep inetd | grep -v -e grep | awk ’{print $2}’‘

8.9 TCP wrapper security

More and more Linux distributions are shipping with secure defaults. But, never ASSUME that things are locked down. CONFIRM IT!

- Edit "/etc/hosts.deny" and insert the following at the end of the file:

```
        ALL: ALL
```

It should also be noted that TCP wrappers supports extensive logging and remote banners. Please see the end of this section for a detailed example.

- edit "/etc/hosts.allow" and insert lines at the end of the file for each IP and or Domain that you want to allow access to the Linux box.

NOTE: Do NOT use DNS names for the hosts as DNS can be spoofed. Use TCP/IP addresses instead.

ALL: 127.0.0.1 #Needed for some local services like comsat

ALL: 200.211.0.40 #Securehost

ALL: w.x.y.z

For example:

```
    ALL:    192.168.0.2     #Allow everything from coyote2
    ALL:    200.211.0.40    #Allow all traffic from Explict Allowed #1
    ALL:    200.211.1.      #Allow *ALL* traffic from all hosts on the 200.211.1.x
                            #network.  Yes, the option should END with a
single "."
```

Or if you want to be more granular, you can do the following. All TCP wrapper supported daemons that you can put in here are noted in the /etc/inetd.conf file.

```
        in.ftpd: 192.168.0.2    #Allow only FTP traffic from coyote2
        in.pop3d: 200.211.0.40  #All only pop-3 traffuc from Explict Allowed #1
```

TCP Wrapper logging and banner support

As mentioned above, TCP wrappers support advanced features like logging and sending text banners to the remote machine. To do this, you want to change the /etc/hosts.deny file to look something like the following:

```
# The following example will DENY all traffic except finger.
#   For finger, it will allow the request but log it, send a banner and THEN
#   deny it
#
# First, set up a booby trap and bounce message for all except finger
# and log attempt to /var/log/tcpwrappers.log

ALL except in.fingerd: ALL \
    :spawn (/usr/sbin/safe_finger -l @%h | /bin/mail -s %d-%h root;\
        date >>/var/log/tcpwrappers.log;\
        echo '%u@%h (%d) connection attempted.' >>/root/access.log)& \
    :rfc931 45\
```

```
    :twist /bin/echo \
        $'\nAccess to this system is limited to authorized users. \
        \n%u@%h is not a valid ID to access %d \
        \non this system. This attempt has been logged. \n'

# Now log and bounce message for finger
#
in.fingerd: ALL\
    :spawn (date >>/var/log/tcpwrappers.log; \
        echo '%u@%h (%d) connection attempted.' >>/var/log/tcpwrappers.log)& \
    :rfc931 45\
    :twist /bin/echo \
        $'\nAccess to this system is limited to authorized users. \
        \n%u@%h is not a valid ID to access %d \
        \non this system. This \
        attempt has been logged.\
        \n'
```

8.10 FTP Anonymous users

Disable anonymous FTP to your box by editing /etc/ftpaccess and change the common first line that looks like:

```
                class   all   real,guest,anonymous  *
```

...to this (notice the words "guest" and "anonymous" is gone:

```
                class   all   real *
```

8.11 Shadow Passwords

In most earily Linux distributions, all user's passwords were stored in the /etc/passwd file. These passwords were then encrypted by the "crypt" tool. The problem with this setup was that anyone could get these encrypted passwords and crypt's encryption was very poor. These passwords could then be broken with publically available tools. In recent times, the shadow system was implemented where the passwords were hashed with the MD5 algorithm and placed the resulting MD5 hased passwords in /etc/shadow.

To quickly see if your machine is "shadow" enabled, look at the "/etc/passwd" file. In this file, you will see the username, password, UserID (UID), GroupID (GID), Home Directory, and the user's default shell all separated by colons (:). Anyway, if you see "x"s in the second left-hand field, the password field, then you are done! If you DON'T see "x"s in that field.. you need to follow these directions or better yet.. get a newer distribution!

8.11.1 Slackware 3.x

Slackware v3.2 did not come with Shadow passwords enabled but v3.4+ does. For several reasons, I recommend that you just upgrade to Slackware v3.4 if you are running an older Slackware distribution. The upgrade will fix numerous security issues and has many other features as well.

8.11.2 Redhat

Redhat5, out of the box, does NOT do shadow passwords (stupid) but it is fixed in RH 6.1 and onward.

Confirm that your system is using SHADOW passwords by looking at the /etc/passwd file and make sure that the second left-hand field next to the username is a ":x:". If so, make sure everthing in this section is setup the same on your box.

If it isn't do the following:

- login as root

- type in "pwconv"

- This will convert the /etc/passwd file and move the encrypted passwords over to /etc/shadow and change the encryption algorithm from the weak "crypt" system to "md5"

- More info is available in "/usr/doc/pam-0.64/txts/pam.txt"

- NOTE: Using passwords more than 8 characters will NOT work. Use larger passwords and prepare NOT to be able to login again!

- Edit the /etc/pam.d/passwd file and change the bottom lines

NOTE: There are (2) methods shown below. Crypt is the OLD UNIX method and is considered weak. The newer method uses MD5 hashing. I recommend the MD5 method.

So, edit the file and change it to the following:

For MD5 hashing (more secure and recommended):

```
--
auth       required     /lib/security/pam_pwdb.so shadow nullok
account    required     /lib/security/pam_pwdb.so
password   required     /lib/security/pam_cracklib.so retry=3
password   required     /lib/security/pam_pwdb.so shadow use_authtok nullok md5
--
```

For normal CRYPT hashing:

```
--
auth       required     /lib/security/pam_pwdb.so shadow nullok
account    required     /lib/security/pam_pwdb.so
password   required     /lib/security/pam_cracklib.so retry=3
password   required     /lib/security/pam_pwdb.so shadow use_authtok nullok
--
```

8.12 Disable ROOT TELNET/SSH access

By default, most Linux distributions don't allow direct "root" logins via TELNET or SSH. This is considered good security.

- If you DO need to login via telnet as root then edit or create the /etc/securetty file and ADD the following:

```
ttyp0
ttyp1
ttyp2
```

Please note that newer Linux distributions now use the DevFS system. If your system uses DevFS, you should add the following in addition to the "ttyp0, ttyp1, etc." system. If you are using DevFS full time, you can delete the ttyp0, etc. lines.

```
vc/1
vc/2
```

**** MAKE SURE YOU PUT "#"s IN FRONT OF THESE NEW LINES ONCE YOU ARE DONE! ****

8.13 Disable ROOT FTP access

It seems that some Linux distributions do not come with the /etc/ftpusers file. This file basically is for when any usernames in this file, they are NOT allowed to FTP in. Usually, it is considered POOR security to be able to FTP in as ROOT. By putting the word "root" into this file, this disables FTP logins from "root".

- If you ever need to FTP into the linux box as ROOT (you shouldn't be able to by default), edit the "/etc/ftpusers" file and put a "#" in front of "root".

NOTE: If the /etc/ftpusers file DOESN'T already exist, just create it. Once you are done, LEAVE it there with at least the line "root" without a "#" in front of it.

```
**********************************************************
**** MAKE SURE YOU REMOVE THIS "#" ONCE YOU ARE DONE ****
****       SINCE THIS IS A BIG SECURITY ISSUE        ****
**********************************************************
```

8.14 Disable miscellaneous cron stuff

* When users install Redhat, they usually install more programs than they plan to initially use. Though Redhat allows users to later choose what daemons are and are NOT run upon boot, this does NOT disable some things that are loaded into the cron file.

As mentioned before in this section, unless you plan on using the functionality of a specific product, DON'T disable a given cron entry. Just delete the package all together as described above.

8.14.1 Redhat users:

NOTE: DON'T disable: logrotate, tmpwatch, updatedb.cron, makewhatis.cron

- Look in the /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly and make sure that nothing is installed that you don't want. For example, I had to do the following for RH 5.2:

```
mkdir -m 700 /etc/cron.disabled
mkdir -m 700 /etc/cron.disabled/cron.hourly
mkdir -m 700 /etc/cron.disabled/cron.daily

mv /etc/cron.hourly/inn-cron-nntpsend /etc/cron.disabled/cron.hourly
mv /etc/cron.daily/inn-cron-expire /etc/cron.disabled/cron.daily
mv /etc/cron.daily/inn-cron-rnews /etc/cron.disabled/cron.daily
mv /etc/cron.daily/tetex.cron /etc/cron.disabled/cron.daily
```

8.14.2 Slackware Users:

NOTE: DON'T disable: updatedb.cron

- Realistically, you won't have the same issues as Redhat users because Slackware doesn't have as many bells and whistles as RH does. BUT, check to make sure. All of Slackware's cron configuration is stored here.

```
less /var/spool/cron/crontabs/root
```

8.15 File Permission corrections

A lot of the default file permissions on Linux distributions just give away too much information to the end user or hacker. Some people might think that some of these are paranoid but I'd rather be safe than sorry:

NOTE: Most of these permissions reflect Redhat 5.2 but most will apply to any Linux distribution.

NOTE2: If you receive any ERRORs when applying these changes, don't worry. That just means you don't have that package installed.

It is highly recommended that you apply these permissions via the TrinityOS-security script to avoid typing mistakes and save time.

```
# Files in /dev
chmod 660 /dev/lp*

# Files in /bin
echo "Bru is a commercial backup program but some Linux distributions come with it"
chmod 750 /bin/bru
chmod 750 /bin/linuxconf
chmod 750 /bin/mount
chmod 750 /bin/mt
chmod 750 /bin/rpm
chmod 750 /bin/setserial
chmod 4750 /bin/su
chgrp adm /bin/su
chmod 750 /bin/umount

# Files in /sbin
chmod 750 /sbin/accton
chmod 750 /sbin/badblocks
chmod 750 /sbin/ctrlaltdel
chmod 750 /sbin/chkconfig
chmod 750 /sbin/chkraid
chmod 750 /sbin/debugfs
chmod 750 /sbin/depmod
chmod 750 /sbin/dhcpcd
chmod 750 /sbin/dump*
chmod 750 /sbin/fdisk
chmod 750 /sbin/fsck*
chmod 750 /sbin/ftl*
chmod 750 /sbin/getty
chmod 750 /sbin/halt
```

```
chmod 750 /sbin/hdparm
chmod 750 /sbin/hwclock
chmod 750 /sbin/ide_info
chmod 750 /sbin/if*
chmod 750 /sbin/init
chmod 750 /sbin/insmod
echo "IPFWADM is only installed for v2.0 kernels"
chmod 750 /sbin/ipfwadm
chmod 750 /sbin/ipx*
chmod 750 /sbin/isapnp
chmod 750 /sbin/kerneld
chmod 750 /sbin/killall*
echo "This is the new location for klogd.  Please disregard any errors if this doesn't work."
chmod 750 /sbin/klogd
chmod 750 /sbin/lilo
chmod 750 /sbin/mgetty
chmod 750 /sbin/mingetty
chmod 750 /sbin/mk*
chmod 750 /sbin/mod*
chmod 750 /sbin/netreport
chmod 750 /sbin/pam*
chmod 750 /sbin/pcinitrd
chmod 750 /sbin/pnpdump
chmod 750 /sbin/portmap
chmod 750 /sbin/quotaon
chmod 750 /sbin/raidadd
chmod 750 /sbin/restore
chmod 750 /sbin/runlevel
chmod 750 /sbin/stinit
echo "This is the old location for klogd.  Please disregard any errors if this doesn't work."
chmod 750 /sbin/syslogd
chmod 750 /sbin/swapon
chmod 750 /sbin/tune2fs
chmod 750 /sbin/uugetty
chmod 750 /sbin/vgetty

echo "Files in /usr/bin"
chmod 750 /usr/bin/control-panel
chmod 750 /usr/bin/comanche
chmod 750 /usr/bin/eject
chmod 750 /usr/bin/glint
chmod 750 /usr/bin/gnome*
chmod 750 /usr/bin/gpasswd
chmod 750 /usr/bin/ipx*
chmod 750 /usr/bin/kernelcfg

chmod 755 /usr/bin/lp*
chmod 4755 /usr/bin/lpr

#NOTE: I feel setting "lpr" to allow any group to execute it is
```

```
#        a bad thing.
#
#        I would like to add UNIX users and even the Samba process to
#        the "lp" group already defined in /etc/groups and then be able
#        to put things back to to 4750.  BUT, I just talked to a buddy
#        of mine and this really isn't possible.  Linux doesn't support
#        multiple groups per file and Linux doesn't support access lists
#        (ACLs') yet.  So, you either have to do all this or run LPRng.
#
#        Stock permissionss are:
#               -r-sr-sr-x    1 root     lp          15436 Oct 17 06:49 lpq
#               -r-sr-sr-x    1 root     lp          16176 Oct 17 06:49 lpr
#               -r-sr-sr-x    1 root     lp          16132 Oct 17 06:49 lprm

chmod 750 /usr/bin/mformat
chmod 750 /usr/bin/minicom
chmod 750 /usr/bin/mtools
chmod 750 /usr/bin/netcfg
chmod 750 /usr/bin/rusers
chmod 750 /usr/bin/rwall
chmod 750 /usr/bin/uucp


echo "Files in /usr/sbin"
chmod 750 /usr/sbin/am*
chmod 750 /usr/sbin/at*
chmod 750 /usr/sbin/automount
chmod 750 /usr/sbin/bootp*
chmod 750 /usr/sbin/crond
chmod 750 /usr/sbin/dhc*
chmod 750 /usr/sbin/dip
chmod 750 /usr/sbin/dump*
chmod 750 /usr/sbin/edquota
chmod 750 /usr/sbin/exportfs
chmod 750 /usr/sbin/fixmount
chmod 750 /usr/sbin/ftpshut
chmod 750 /usr/sbin/gated
chmod 750 /usr/sbin/group*
chmod 750 /usr/sbin/grp*
chmod 750 /usr/sbin/imapd
chmod 750 /usr/sbin/in.*
chmod 750 /usr/sbin/inetd
chmod 750 /usr/sbin/ipop*
echo "This is the old location for klogd.  Please disregard any errors if this doesn't work."
chmod 750 /usr/sbin/klogd
chmod 750 /usr/sbin/logrotate
chmod 750 /usr/sbin/lp*
chmod 755 /usr/sbin/lsof
chmod 750 /usr/sbin/makemap
chmod 750 /usr/sbin/mk-amd-map
```

```
chmod 750 /usr/sbin/mouseconfig
chmod 750 /usr/sbin/named*
chmod 750 /usr/sbin/nmbd
chmod 750 /usr/sbin/newusers
chmod 750 /usr/sbin/ntp*
chmod 750 /usr/sbin/ntsysv
chmod 750 /usr/sbin/pppd
chmod 750 /usr/sbin/pnpprobe
chmod 750 /usr/sbin/pw*
chmod 750 /usr/sbin/quota*
chmod 750 /usr/sbin/rdev
chmod 750 /usr/sbin/rdist
chmod 750 /usr/sbin/repquota
chmod 750 /usr/sbin/rhbackup
chmod 750 /usr/sbin/rotatelogs
chmod 750 /usr/sbin/rpc*
chmod 750 /usr/sbin/rwhod
chmod 750 /usr/sbin/samba
chmod 750 /usr/sbin/setup
chmod 750 /usr/sbin/showmount
chmod 750 /usr/sbin/smb*
chmod 750 /usr/sbin/sndconfig
chmod 750 /usr/sbin/snmp*
chmod 750 /usr/sbin/squid
echo "This is the old location for sysklogd.  Please disregard any errors if this doesn't work."
chmod 750 /usr/sbin/syslogd
chmod 750 /usr/sbin/taper
chmod 750 /usr/sbin/tcpd*
chmod 750 /usr/sbin/time*
chmod 750 /usr/sbin/tmpwatch
chmod 750 /usr/sbin/tunelp
chmod 750 /usr/sbin/user*
chmod 750 /usr/sbin/uu*
chmod 750 /usr/sbin/vi*
chmod 750 /usr/sbin/wire-test
chmod 750 /usr/sbin/xntp*
```

8.16 SUID ROOT PROGRAMS

- Check that there aren't any SUID ROOT (programs that execute as the ROOT user) that are WRITABLE by other users. To do this, execute this following command (per `<http://rlz.ne.mediaone.net/linux/index.html>`):

```
                mkdir -m700 /etc/info
                find / -type f \( -perm -04000 -o -perm -02000 \) -ls > /etc/info/suid-results
```

So what do you do with these results?

Figure out the SUID programs that you need and note which ones they are and where they are. The issue is to just make sure that no other unknonwn programs don't get added to this list. What about just changing their permissions to NOT be SUID root? This would be bad because most programs that are usually SUID ROOT *must* be this way or they won't work right.

But, for example, GnuPlot on a recent copy of SuSE was found SUID though it shouldn't have been. Later, a person on BugTraq found this and created both a root exploit and patch for it. So, this is where you can be proactive and fix things.

For the other SUID programs you don't need or know what they are, change their permissions to 700 (chmod 700 *) or even better yet, change their permissionss to 700, move them to a temporary directory to later delete them once you are SURE you don't need the programs.

*** Once you have resolved all your SUID issues, rename this *** /etc/info/suid-results file to /etc/info/suid-results-checked and then *** fix the permissions:

```
mv /etc/info/suid-results /etc/info/suid-results-checked
chmod 600 /etc/info/suid-results-checked
```

We will use this file later as a template file to check for changed SUID files in 9 (Section 9)

8.17 Looking for R-command files

Much like looking for SUID files above, it is also a good idea to look for R-command permission files.

```
find / | grep -e ".rhosts" -e "hosts.equiv" > /etc/info/rcmd-results
```

Once you have reviewed this /etc/info/rcmd-results file for any entries that DON'T belong in there, rename it and fix its permissions:

```
mv /etc/info/rcmd-results /etc/info/rcmd-results-checked
chmod 600 /etc/info/rcmd-results-checked
```

8.18 Fix Xwindows permissions

* This was exploited recently in Xfree86 but I still feel that the sticky bit on the /tmp/.X11-unix directory should be set

```
rm -rf /tmp/.X11-unix
mkdir -p -m 1777 /tmp/.X11-unix
chmod o+t /tmp/.X11-unix
```

9 Advanced System Logging and some Cool Tips

9.1 SYSLOG tuning

- SYSLOG is the main UNIX logging tool. With this system, you can setup logging to be very high level to extremely detailed and have each logging stream go to a different file. Trust me, SYSLOG is your friend!

Edit /etc/syslog.conf and -ADD- the following lines if they aren't already in there:

******* * NOTE!!! All space from the left and right columns MUST BE TABS. * If they are SPACEs, syslog will NOT load! Kinda stupid eh? *

Redhat users:

```
*.warn;*.err                                    /var/log/syslog
auth.*;user.*;daemon.none                       /var/log/loginlog
kern.*                                          /var/log/kernel
```

Slackware users:

```
*.warn;*.err                                    /var/adm/syslog
mail.*                                          /var/adm/maillog
auth.*;user.*;daemon.none                       /var/adm/loginlog
kern.*                                          /var/adm/kernel
```

All Distributions: Once you have edited the /etc/syslog.conf file, save your changes and exit the editor. Now, following files must be created for SYSLOG to work:

```
touch /var/log/syslog
touch /var/log/loginlog
touch /var/log/kernel
```

Next, you might see in your /var/log/messages and /var/log/syslog files lines that look like:

```
--
Nov 28 08:25:42 hostname -- MARK --
--
```

This is the SYSLOG daemon telling you that SYSLOG is running but had nothing to report. If you don't like this behavior, you can disable it by editing the following file and changing the MARK time out.

In /etc/rc.d/init.d/syslog, find the line that says:

```
--
daemon syslogd
--
```

and replace it with:

```
--
daemon syslogd -m 0
--
```

To make ALL of the above changes go into effect, run:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}’‘

Next, close down these new files (and existing files) permissions:

9.1.1 Redhat:

```
chmod 600 /var/log/syslog
chmod 600 /var/log/loginlog
chmod 600 /var/log/kernel
echo "Make sure old SYSLOG file perms are ok too."
chmod 600 /etc/syslog.conf
chmod 600 /var/log/cron
chmod 700 /var/log/httpd
chmod 600 /var/log/httpd/*
chmod 600 /var/log/maillog
chmod 600 /var/log/messages
chmod 600 /var/log/mysql
chmod 600 /var/log/netconf.log
chmod 700 /var/log/samba
chmod 600 /var/log/samba/*
chmod 600 /var/log/sendmail.st
chmod 600 /var/log/secure
chmod 600 /var/log/spooler
chmod 700 /var/log/squid
chmod 600 /var/log/squid/*
chmod 600 /var/log/xferlog
```

9.1.2 Slackware:

```
chmod 600 /var/adm/syslog
chmod 600 /var/adm/loginlog
chmod 600 /var/adm/kernel
chmod 600 /etc/syslog.conf
```

Ok, now restart SYSLOG:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}“

9.2 Log Rotations

Stock Redhat comes with a tool that will take your SYSLOG log files, rename them to the day they came from, optionally compress them, and then restart the log files for the next day. This is very handy as SYSLOG files can get VERY large. If you are using some other Linux distribution that doesn't have this feature, I highly recommend installed a program that will do this for you (there are many to choose from).

- Redhat:

Next, allow the new syslog file to be rotated as well. Add these lines to the /etc/logrotate.d/syslog:

```
--
/var/log/kernel {
        postrotate
        /usr/bin/killall -9 klogd
      /sbin/klogd &
```

```
        endscript
}

/var/log/loginlog {
        postrotate
        /usr/bin/killall -HUP syslogd
        endscript
}

/var/log/syslog {
        postrotate
        /usr/bin/killall -HUP syslogd
        endscript
}
--
```

Also.. I highly recommend that you edit the /etc/logrotate.conf file and do the following:

Find "#compress" and remove the "#" so it only says "compress".

I also recommend that your #ed out the sections to look like this:

[Why? If these files are rotated, you won't be easily able to] [tell when users have logged in.]

```
## no packages own lastlog or wtmp -- we'll rotate them here
#/var/log/wtmp {
#    monthly
#    rotate 1
#}

#/var/log/lastlog {
#    monthly
#    rotate 1
#}
```

This will then compress the moved log files with Gzip.

Finally, some log files explicitly default to no-compression. Why? I recommend to add a "#" before the "nocompress" line in each of the following files:

```
/etc/logrotate.d/ftpd
/etc/logrotate.d/linuxconf
/etc/logrotate.d/sendfax
```

There might be other files in this directory. Check each one of them.

Lastly, I recommend to go into the /etc/logrotate.d/ directory and MOVE log config files that you KNOW you won't be using to a "disabled" directory. This is completely dependant on the services that you installed and then on which ones you opted to NOT run.

As mentioned before, for packages that you KNOW you won't ever use, instead of disabling the logrotation for a given package, DELETE the entire package either using RPM or PKGDEL.

To manually disable things:

```
mkdir -m 700 /etc/logrotate.d.disabled
mv /etc/logrotate.d/mysql /etc/logrotate.d.disabled
mv /etc/logrotate.d/squid /etc/logrotate.d.disabled
```

9.3 Cool rc.local tips and LOGIT for logging troubleshooting

- Edit the "/etc/rc.d/rc.local" file and add the following lines at the end:

The following tip is a personal idea I like for both Redhat and Slackware. By default, then you login to a Linux box, it tells you the Linux distribution name, version, kernel version, and the name of the server. Even worse, Mandrake puts up a very stupid looking Penguin.

To me, this is giving away too much info. I rather just prompt users with a "Login: " prompt (if they ever get that far past your packet firewall and TCP wrappers).

To fix this, do the following:

Place "#"s in front of the following lines like shown:

NOTE: This looks a little different with Mandrake:

/etc/rc.d/rc.local

```
## This will overwrite /etc/issue at every boot.  So, make any changes you
## want to make to /etc/issue here or you will lose them when you reboot.
#echo "" > /etc/issue
#echo "Red Hat Linux $R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
#
#cp -f /etc/issue /etc/issue.net
```

Then, do the following:

```
- rm -f /etc/issue
- rm -f /etc/issue.net
- touch /etc/issue
- touch /etc/issue.net
- chmod 400 /etc/issue
- chmod 400 /etc/issue.net
```

Also, if your Linux box stays up for several months, any kernel messages, errors, firewall hits, etc will OVERWRITE the output from "dmesg". Personally, I *HATE* this but my work-around is to make a "dmesg" copy upon every boot. Append the following to the bottom of your /etc/rc.d/rc.local file:

/etc/rc.d/rc.local

```
dmesg >> /etc/info/dmesg
```

* Next, the following tip is a great way of seeing your various logs on your Linux box without having to login, etc. Some people might feel that this is a security risk but the risk stems from physical security.

Edit the following file and FIND each line for, say syslog or messages, and add in the respective line:

/etc/syslog.conf

```
*.warn;*.err                                    /dev/tty7
mail.*                                          /dev/tty8
kern.*                                          /dev/tty8
```

To make these changes take effect, run the following line:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}’‘

Now, whenever anything is added to those log files, just go to the ALT-F7 or F8 VTY and see the messages roll by in real-time.

* Like the real-time log monitor above, it’s nice to be able to see errors in real time whenever you suspect problems via a TELNET, SSH, etc. To do this, create the file with the following:

Slackware:

/root/logit

```
--
#/bin/sh
tail -f /var/adm/samba/log.nmb &
tail -f /var/adm/samba/log.smb &
tail -f /var/adm/xferlog &
tail -f /var/adm/maillog &
tail -f /var/adm/secure &
tail -f /var/adm/syslog &
tail -f /var/adm/messages &
--
```

Redhat:

/root/logit

```
--
#!/bin/sh
tail -f /var/log/samba/log.nmb &
tail -f /var/log/samba/log.smb &
tail -f /var/log/xferlog &
tail -f /var/log/maillog &
tail -f /var/log/secure &
tail -f /var/log/syslog &
tail -f /var/log/messages &
--
```

Now, fix the permissions for it:

chmod 700 /root/logit

Close the file and then fix it’s permissions with "chmod 700 /usr/local/sbin/logit".

Now, whenever you are suspecting problems with ANYTHING on your Linux box, just run "/root/logit" and watch the error logs go by in real-time.

A few tips: - type in "clear" at the UNIX prompt now and then to clean the screen up for readibility sake.

- When logs are scrolling by but you are looking for something that should show up in a few seconds, hit ENTER a few times to move up the old log info a few lines.

When you are done with "logit", run the command "killall tail" to stop all the logging.

9.4 A more readable BASH prompt

Being a command line junky, I use the CLI (command line interface) most of the time. To make things a little easier on the eye, I recommend that you make the BASH prompt a little more easy on the eye. All NON-root users will get a "green" colored prompt but ROOT users will get a "red" colored prompt.

You can do this one of two ways. Have it setup on a PER USER basis or for ALL users.

For this example, let's do it just for the ROOT user.

1. Copy the main bash profile to the root user's home directory:

```
cp /etc/bashrc /root/.bashrc
```

NOTE: Why bashrc and not profile? The reason being is that bashrc OVERRIDES anything in the profile.

2. Edit it and find the line for the "PS1" variable and REPLACE it with the following. This will make the prompt be a bright green (easy on the eyes) color for NON-root users and red for ROOT uses. It will also show the machine name and a condensed directory prompt:

```
if [ ‘id -un‘ = root ]; then
    PS1=’\[\033[1;31m\]\h:\w\$\[\033[0m\] ’
else
    PS1=’\[\033[1;32m\]\h:\w\$\[\033[0m\] ’
fi
```

---

3. Save the .bashrc, login as the root user or run "su -" and then you should have the new prompt. For more good Bash ideas, check out the BASH howto from 5 (Section 5).

If you wanted to do it for ALL users, do the above changed to the /etc/bashrc file.

9.5 Some security tips for BASH

As you execute commands in bash, they are recorded for the command history, etc. Though this is great during your shell login, you might accidently put a password in as a command, etc. To clean this up and cover your tracks once you log off, add the following line as the LAST line in your /etc/profile:

```
/etc/profile
--<begin>
#Depending on your version of BASH, you might have to use
# the other form of this command
        trap "rm -f ~$LOGNAME/.bash_history" 0

#The older KSH-style form
        trap 0 rm -f $LOGNAME/.bash_history
--<end>
```

9.6 Make the apropos database

One powerful command in UNIX is the "apropos" or "man -k" command. This will let you do command searches on generic words like "modem", etc. BUT, when you first install Linux, this database isn't complete. It is usually run as a weekly cron job but I recommend to start it now:

```
        makewhatis -w &
```

NOTE: This command will take a while depending on HD and CPU speed.

If you get ERRORs on the "makewhatis" command as I did in Mandrake 6.1, some of this is how to fix them. I received the following errors (bugs in the distribution - already reported as Bug #ier206). Running this command in Mandrake 7.0 runs without error.

```
--
bzcat: Can't open input file ./fetchmailconf.1.bz2: No such file or directory.
bzcat: ./ksh.1.bz2 is not a bzip2 file.
bzcat: Can't open input file ./pdksh.1.bz2: No such file or directory.
Read file error: ./rec.1 No such file or directory
bzcat: ./tixwish.1.bz2 is not a bzip2 file.
bzcat: ./efence.3.bz2 is not a bzip2 file.
Read file error: ./stm.8 No such file or directory
Read file error: ./clockprobe.8 No such file or directory
--
```

---

line 1: The /usr/man/man1/fetchmailconf.1.bz2 file is a symbolic link to fetchmail.1. This file doesn't exist since its compressed with bz2. To fix it, do:

```
rm /usr/man/man1/fetchmailconf.1.bz2
ln -s /usr/man/man1/fetchmail.1.bz2 /usr/man/man1/fetchmailconf.1.bz2
```

---

line 2: The /usr/man/man1/ksh.1.bz2 file isn't really bz2'ed. To fix it, do:

```
mv /usr/man/man1/ksh.1.bz2 /usr/man/man1/ksh.1
bzip2 -z /usr/man/man1/ksh.1
```

---

line 3: The /usr/man/man1/pdksh.1.bz2 file points to a non-bz2 file. (sloppy). To fix it, do:

Do the line-2 fix above

```
rm /usr/man/man1/pdksh.1.bz2
ln -s /usr/man/man1/ksh.1.bz2 /usr/man/man1/pdksh.1.bz2
```

---

line 4: The /usr/man/man1/rec.1 file points to a bogus path /var/tmp/sox-root//usr/man/man1/play.1 (sloppy). To fix it, do:

```
rm /usr/man/man1/rec.1
ln -s /usr/man/man1/play.1.bz2 /usr/man/man1/rec.1.bz2
```

---

line 5: The /usr/man/man1/tixwish.1.bz2 file is not a bz2 file. To fix it, do:

```
mv /usr/man/man1/tixwish.1.bz2 /usr/man/man1/tixwish.1
bzip2 -z /usr/man/man1/tixwish.1
```

line 6: The /usr/man/man3/efence.3.bz2 file is not a valid man page To fix it, do:

```
rm /usr/man/man3/efence.3.bz2
```

line 7: The /usr/man/man8/stm.8 file points to a non existing file. To fix it, do:

```
rm /usr/man/man8/stm.8
ln -s /usr/man/man8/SVGATextMode.8.bz2 /usr/man/man8/stm.8.bz2
```

line 8: The /usr/man/man8/clockprobe.8 file points to a non existing file. To fix it, do:

```
rm /usr/man/man8/clockprobe.8
ln -s /usr/man/man8/grabmode.8.bz2 /usr/man/man8/clockprobe.8.bz2
```

Once you have fixed these problems, re-run "makewhatis -w" and make sure it completes cleanly.

9.7 Sendlogs - Daily email of system logs with log reduction

** HIGHLY RECOMMENDEDD for ALL Administrators **

If you are like me, you would like to know if any strange things are happening to your system like (processes failing, hacker attempts, etc.). At the same time, you probably don't have the time to scan over all these logs every day to see what is and isn't interesting. This script will simply count the number of specific blocked port connections (worms, viruses, etc.). This script also optionally monitors how many times your modem line came online (or failed due to busy signals, etc.) and report what speeds it connected at in a nice summarized table.

To do this, follow these next steps (note: this isn't the prettiest script I've wrote and it needs a LOT of cleaning but it should work for you).

*** Note:

- Other tools like Psionic LogCheck and Stanford's Swatch tools do similar things but in in a MUCH cleaner fashion. As I get get those solutions running, this script will be replaced.

```
ALL USERS:              The first time this script executes, you
                        will receive some errors regarding:

                                - todays-date and yesterdays-date

                        You can safely ignore these errors!



Slackware users:        This file should be called "/usr/local/sbin/sendlogs"

Redhat users:   This file should be called "/usr/local/sbin/sendlogs"
```

```
                (Note:  All users:  you will need to substitute in your proper mail address
                (                               so you will get your logs
                (
                (         Slackware users: please edit this file and change the /var/log
                (                       references to /var/adm
                (
                (       Modem users:  You will need to un-# out the modem fields and
                (                               make sure that the temp file swaping from
                (                               $1.tmp to $2.tmp etc. transisions are correct.
                (
                (                               I have this disabled because I'm a cable modem dude
                (                               now but this worked well.
```

All of TrinityOS's step-by-step instructions, files, and scripts are fully scripted out for an automatic installation at:

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

/usr/local/sbin/sendlogs <Sendlogs START>

```
#!/bin/sh

# TrinityOS-sendlogs.sh

# 03/06/04
#
# Part of the copyrighted and trademarked TrinityOS document.
# <"http://www.ecst.csuchico.edu/~dranch">
#
# Written and Maintained by David A. Ranch
# dranch@trinnet.net
#
# Updates:
#
# 03/06/04 - Added counts for SQL
# 02/12/04 - Added counts for MyDoom trojans
# 01/12/04 - Added Samba counts to the DMZ segment
# 11/15/03 - Fixed a typo of > vs. >> for the cups and http filter
# 11/09/03 - added a count of port 631 hits (CUPS)
# 10/28/03 - Changed mirror DD drive to sdc
# 10/23/03 - Adding a logger debug command
# 09/26/03 - Added a count of port 80 hits (www)
# 09/23/03 - removed all port 80 hits
# 01/30/03 - Added MP3 archive change log
# 06/28/02 - Added Seti stats
# 12/13/01 - Added a calculated total runtime to the end of the script
# 11/13/01 - filter those damn run-parts messages
# 08/28/01 - Log the status of the script for debuging hangs
# 07/14/01 - delete all the Jeff R denied update messages
```

```
# 01/07/01 - This script is now parsed directly from the SGML code and
#            because of this, several formatting issues were fixed.
#          - Made the output a little more pretty
#          - #ed out some diagnostic file information
#          - added an lsof log entry
#          - cleaned up the error reports in the SUID and RCMD searches
#
# 12/26/00 - Added --MARK-- Filtering
#
# 10/28/00 - Added an optional and #ed out section on DDing one HD to
#            another.  This is a simple but VERY effective online backup
#            though it is only done once a night.  If you have a spare HD
#            in your system, this is the next best thing to setting up
#            RAID1.  Personally, I just recommend to setup RAID1!  :)
#
# 10/08/00 - Deleted the removal of the SUID and RCMD new result files
#
# 09/16/00 - Added a full RPM database verification setup
#
# 04/15/00 - Added the $HOST variable to easily tune the SUBJECT field to
#            reflect the name of your Linux system.  You should edit this
#            to reflect your system.
#
# 04/09/00 - Hmmm.. we need %e and NOT %d for catching dates 01-09.
#            Basically, I need to reverve the change on 01/17/00.
#
# 02/21/00 - Doh!  We do need the spaces between %b and %d
#
# 01/17/00 - Fixed all the "date" issues.  Date now uses %d over %e and
#            doesn't use any spaces.
#
# 01/01/00 - Fixed a missing ">" on line 139
#
# 12/16/99 - Fixed the RCMD mailer command at the end.  The "mail -s" line
#            needed to be ONE line
#
# 11/26/99 - Cleaned things up a bit
#          - Made all file references absolute
#
# 02/01/99 - Added "w" to the vitals output

logger "Sendlogs starting: `date`"

# Change this variable to reflect the HOSTNAME of this box
# -------------------------------------------------------
HOST="roadrunner"
EXTIP="100.200.0.212"

export COLUMNS=132
```

```
echo "Sendlogs start: ‘date‘" > /var/log/sendlogs.status
START=‘date +%s‘


#Make sure that the "yesterdays-date" file exists.  If not, create it.
#
if [ -f /var/log/todays-date ]; then
     mv /var/log/todays-date /var/log/yesterdays-date;
  else
     date +’%b %e’ > /var/log/yesterdays-date;
fi


#Make sure that the "/etc/info/logs" directory exists.  If not, create it.
#
if [ -a /etc/info ]; then
  if [ -a /etc/info/logs ]; then
       echo "";
    else
       mkdir /etc/info/logs;
  fi
  else
     mkdir /etc/info;
     mkdir /etc/info/logs;
fi


date +’%b %e’ > /var/log/todays-date

echo "   Start messages: ‘date‘" >> /var/log/sendlogs.status
cat /var/log/messages | grep "‘cat /var/log/yesterdays-date‘" > /var/log/messlog.‘date +’%b%d%y’‘
export f1=/var/log/messlog.‘date +’%b%d%y’‘
export f2=/var/log/testfile
#echo "File 1: $f1"
#echo "File 2: $f2"

#For messages - FTP and PPP stuff
#
sed -e "/PWD/d" -e "/PASV/d" -e "/TYPE/d" -e "/PORT/d" -e "/NLST/d" -e "/SYST/d" $f1 > $f1.tmp
sed -e "/PASS/d" -e "/QUIT/d" -e "/LIST/d" -e "/CDUP/d" -e "/ATDT/d" -e "/Welcome/d" $f1.tmp > $f2.tmp
sed -e "/Using/d" -e "/Connect/d" -e "/Remote/d" -e "/IP address/d" -e "/CHECKSUM/d"  $f2.tmp > $f1.tmp
sed -e "/Terminated/d" -e "/Terminating/d" -e "/diald/d" -e "/2.2.0/d" -e "/Exit./d" $f1.tmp > $f2.tmp
sed -e "/(passwd=guest)/d" -e "/alarm/d" -e "/Failed/d" $f2.tmp > $f1.tmp

#For messages - modem specific stuff
#
#sed -e "/send /d" -e "/expect/d" -e "/OK/d" -e "/AT&F/d" -e "/ATZ/d" -e "/ ^M /d" $f1.tmp > $f2.tmp
#sed -e "/Swansea/d" -e "/logging/d" -e "/starting/d" -e "/Ready/d" -e "/0x03f8/d" -e "/0x02f8/d" $f2.tmp > $f1.t
#sed -e "/sbpcd.c/d" -e "/CR-563/d" -e "/copyright/d" -e "/sockets/d" -e "/Serial/d" -e "/registered/d" $f1.tmp >
#sed -e "/SLIP/d" -e "/sbpcd-0/d" -e "/ATM0X7/d" -e "/1.44M/d" -e "/8272A/d" -e "/statistics/d" $f2.tmp > $f1.tmp
```

```
#sed -e "/Please/d" -e "/hangup/d" -e "/ip-down/d" -e "/scans/d" $1.tmp -e "/abort on/d" $f1.tmp > $f2.tmp
#sed -e "/CONNECT /d" -e "/BUSY/d" -e "/SIGHUP/d" $f2.tmp > $f1.tmp

#For messages - modem dialout specific stuff
#
#echo -e "---------------------------------------" > /var/log/header.tmp
#echo -e "$HOST Call stats for \c" >> /var/log/header.tmp
#date >> /var/log/header.tmp
#echo -e "                                       " >> /var/log/header.tmp
#echo -e "Total number of connects: \c" >> /var/log/header.tmp
#grep -c "CONNECT" $f1.tmp >> /var/log/header.tmp
#echo -e "      21600: \c" >> /var/log/header.tmp
#grep -c "21600" $f1.tmp >> /var/log/header.tmp
#echo -e "      26400: \c" >> /var/log/header.tmp
#grep -c "26400" $f1.tmp >> /var/log/header.tmp
#echo -e "      28800: \c" >> /var/log/header.tmp
#grep -c "28800" $f1.tmp >> /var/log/header.tmp
#echo -e "      31200: \c" >> /var/log/header.tmp
#grep -c "31200" $f1.tmp >> /var/log/header.tmp
#echo -e "      33600: \c" >> /var/log/header.tmp
#grep -c "33600" $f1.tmp >> /var/log/header.tmp
#echo -e "      33600: \c" >> /var/log/header.tmp
#grep -c "41333" $f1.tmp >> /var/log/header.tmp
#echo -e "      41333: \c" >> /var/log/header.tmp
#grep -c "42666" $f1.tmp >> /var/log/header.tmp
#echo -e "      42666: \c" >> /var/log/header.tmp
#echo -e "                                       " >> /var/log/header.tmp
#echo -e "Total number of busys: \c" >> /var/log/header.tmp
#grep -c "BUSY" $f1.tmp >> /var/log/header.tmp
#echo -e "---------------------------------------" >> /var/log/header.tmp
#echo -e "                                       " >> /var/log/header.tmp
#cat /var/log/header.tmp >> $f1.tmp

#For messages - named specific stuff
#
sed -e "/Cleaned/d" -e "/USAGE/d" -e "/NSTATS/d" -e "/XSTATS/d" $f1.tmp > $f2.tmp
sed -e "/points/d" -e "/Lame server/d" $f2.tmp > $f1.tmp

#For messges - SSH specific
sed -e "/Generating /d" -e "/generation /d" -e "/NSTATS/d" -e "/XSTATS/d" $f1.tmp > $f2.tmp

#For messges - Delete --MARK-- entries and J.Robinson DNS issues
sed -e "/-- MARK --/d" -e "/run-parts/d" $f2.tmp > $f1.tmp

#
# COUNT log hits but delete them -- greatly cuts down on log sizes
#
#

echo -e "Firewall hit log reduction section:" >> /var/log/messlog.tmp
```

```
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp


# --- EXT interfaces ---

#For messages - count all port 80 hits
echo -en "    | Port 80 (www) count: " >> /var/log/messlog.tmp
grep -c "$EXTIP:80" $f1.tmp >> /var/log/messlog.tmp
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 80 stuff
sed -e "/$EXTIP:80/d" $f1.tmp > $f2.tmp

#For messages - count all port 1433 - SQL hits
echo -en "    | Port 1433 (SQL) count: " >> /var/log/messlog.tmp
grep -c "$EXTIP:1433" $f2.tmp >> /var/log/messlog.tmp
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 1443 stuff
sed -e "/$EXTIP:1433/d" $f2.tmp > $f1.tmp

#For messages - count all port 3127 hits
echo -en "    | Port 3127 (MyDoom) count: " >> /var/log/messlog.tmp
grep -c "$EXTIP:3127" $f1.tmp >> /var/log/messlog.tmp
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 3127 stuff
sed -e "/$EXTIP:3127/d" $f1.tmp > $f2.tmp


# --- INT2 interfaces ---

#For messages - count all port 631 hits
echo -en "    | Port 631 (CUPS) count: " >> /var/log/messlog.tmp
grep -c "$INT2BROAD:631" $f2.tmp >> /var/log/messlog.tmp
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 631 stuff
sed -e "/$INT2BROAD:631/d" $f2.tmp > $f1.tmp

#For messages - count all port port 137 hits
echo -en "    | Port 137 (Samba) count: " >> /var/log/messlog.tmp
grep -c "$INT2BROAD:137" $f1.tmp >> /var/log/messlog.tmp
echo -e "    +----------------------------------------------------------" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 137 stuff
sed -e "/$INT2BROAD:137/d" $f1.tmp > $f2.tmp

#For messages - count all port port 138 hits
```

```
echo -en "    | Port 138 (Samba) count: " >> /var/log/messlog.tmp
grep -c "$INT2BROAD:138" $f2.tmp >> /var/log/messlog.tmp
echo -e "    +--------------------------------------------------------\n" >> \
 /var/log/messlog.tmp
#For messges - Delete all PORT 138 stuff
sed -e "/$INT2BROAD:138/d" $f2.tmp > $f1.tmp


mv /var/log/messlog.tmp $f1
cat $f1.tmp >> $f1
#cat $f2.tmp >> $f1
rm -R /var/log/*.tmp

mail -s "$HOST messages for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/messlog.‘date +’%b%d%y’‘

rm /var/log/messlog.‘date +’%b%d%y’‘

echo -e "-------------------------------------------------------"
echo -e "MESSAGES: Parsed, filtered, mailed and deleted messages"
echo -e "-------------------------------------------------------"

#---------------------------------------------

echo "   Start syslog: ‘date‘" >> /var/log/sendlogs.status
cat /var/log/syslog | grep "‘cat /var/log/yesterdays-date‘" > /var/log/syslog.‘date +’%b%d%y’‘

export f1=/var/log/syslog.‘date +’%b%d%y’‘
#echo "file 1: $f1"
#echo "file 2: $f2"

#Syslog - modem specific
#sed -e "/ got /d" -e "/abort on/d" -e "/expect/d" -e "/ ^M /d" -e "/AT&F1^M^M/d" $f1 > $f1.tmp
#sed -e "/ATZ^M^M/d" -e "/ATM0X7S11=40^M^M/d" -e "/Executed/d" -e "/ATDT/d" $f1.tmp > $f2.tmp
#sed -e "/Welcome/d" -e "/Using/d" -e "/Connect/d" -e "/Remote/d" -e "/IP address/d" $f2.tmp > $f1.tmp
#sed -e "/CHECKSUM/d" -e "/Terminated/d" -e "/Terminating/d" -e "/diald/d" -e "/2.2.0/d" $f1.tmp > $f2.tmp
#sed -e "/Exit./d" -e "/(passwd=guest)/d" -e "/alarm/d" -e "/Failed/d" -e "/CONNECT/d" $f2.tmp > $f1.tmp
#sed -e "/hangup/d" -e "/RINGING^M/d" $f1.tmp > $f2.tmp
#mv $f2.tmp $f1

#syslog FTP,
sed -e "/PWD/d" -e "/PASV/d" -e "/LIST/d" -e "/CDUP/d" -e "/RETR/d" -e "/CWD/d" $f1 > $f1.tmp
sed -e "/TYPE/d" -e "/PASS/d" -e "/QUIT/d" $f1.tmp > $f2.tmp

#For messages
sed -e "/send /d" -e "/expect/d" -e "/OK/d" -e "/AT&F/d" -e "/ATZ/d" -e "/ ^M /d" $f2.tmp > $f1.tmp
sed -e "/Swansea/d" -e "/logging/d" -e "/starting/d" -e "/Ready/d" -e "/0x03f8/d" $f1.tmp > $f2.tmp
sed -e "/0x02f8/d" -e "/sbpcd.c/d" -e "/CR-563/d" -e "/copyright/d" -e "/sockets/d" $f2.tmp > $f1.tmp
sed -e "/SLIP/d" -e "/sbpcd-0/d" -e "/1.44M/d" -e "/8272A/d" -e "/statistics/d" $f1.tmp > $f2.tmp
sed -e "/Please/d" -e "/hangup/d" -e "/ip-down/d" -e "/scans/d" $f2.tmp > $f1.tmp
sed -e "/abort on/d" -e "/Serial/d" -e "/registered/d" $f1.tmp > $f2.tmp
```

```
mv $f2.tmp $f1
rm -r /var/log/*.tmp

mail -s "$HOST syslog for `cat /var/log/yesterdays-date`" root@localhost < /var/log/syslog.`date +'%b%d%y'`
rm /var/log/syslog.`date +'%b%d%y'`

echo -e "SYSLOG: Parsed, filtered, mailed and deleted syslog"
echo -e "--------------------------------------------------"


echo "   Start secure: `date`" >> /var/log/sendlogs.status
cat /var/log/secure | grep "`cat /var/log/yesterdays-date`" > /var/log/secure.`date +'%b%d%y'`

export f1=/var/log/secure.`date +'%b%d%y'`
#echo "file 1: $f1"
#echo "file 2: $f2"

sed -e "/127/d" $f1 > $f1.tmp
mv $f1.tmp /var/log/secure.`date +'%b%d%y'`
mail -s "$HOST secure for `cat /var/log/yesterdays-date`" root@localhost < /var/log/secure.`date +'%b%d%y'`
rm -r /var/log/*.tmp 2> /dev/null > /dev/null
rm /var/log/secure.`date +'%b%d%y'`

echo -e "SECURE: Parsed, filtered, mailed and deleted secure"
echo -e "--------------------------------------------------"


echo "   Start xferlog: `date`" >> /var/log/sendlogs.status
cat /var/log/xferlog | grep "`cat /var/log/yesterdays-date`" > /var/log/xferlog.`date +'%b%d%y'`

mail -s "$HOST xferlog for `cat /var/log/yesterdays-date`" root@localhost < /var/log/xferlog.`date +'%b%d%y'`
rm /var/log/xferlog.`date +'%b%d%y'`

echo -e "XFERLOG: Parsed, filtered, mailed and deleted xferlog"
echo -e "----------------------------------------------------"


echo "   Start kernel: `date`" >> /var/log/sendlogs.status
cat /var/log/kernel | grep "`cat /var/log/yesterdays-date`" > /var/log/kernel.`date +'%b%d%y'`

export f1=/var/log/kernel.`date +'%b%d%y'`
export f2=/var/log/testfile

#For kernel - Delete all PORT 80 stuff
sed -e "/$EXTIP:80/d" $f1 > $f1.tmp

mail -s "$HOST kernel for `cat /var/log/yesterdays-date`" root@localhost < /var/log/$f1.tmp

rm -r /var/log/*.tmp 2> /dev/null > /dev/null
```

```
rm /var/log/kernel.‘date +’%b%d%y’‘

echo -e "KERNEL: Parsed, filtered, mailed and deleted kernel"
echo -e "---------------------------------------------------"


echo "   Start vitals: ‘date‘" >> /var/log/sendlogs.status

df > /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
w >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
free >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
ps aux >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
lsof -i >> /var/log/sendlogs.‘date +’%b%d%y’‘

mail -s "$HOST vitals for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/sendlogs.‘date +’%b%d%y’‘
rm -f /var/log/sendlogs.‘date +’%b%d%y’‘

echo -e "VITALS: Sent system vitals.."
echo -e "----------------------------"


# Create a full file system ls-laR archive in /etc/info
#
#  NOTE:  You should ALSO copy this file to somewhere on a DIFFERENT HD,
#  floppy, etc. in case your mail HD fails.
#
echo "   Start ls-laR: ‘date‘" >> /var/log/sendlogs.status
ls -laR / 2> /dev/null | bzip2 -9 > /etc/info/logs/ls-laR.‘date +’%b%d%y’‘.bz2
echo -e "LS-LAR: Created full file system ls-laR archive in /etc/info"
echo -e "------------------------------------------------------------"
# cp /etc/info/logs/ls-laR.‘date +’%b%d%y’‘.bz2 /to/some/other/HD


# Create a full file system du archive in /etc/info
#
#  NOTE:  You should ALSO copy this file to somewhere on a DIFFERENT HD,
#  floppy, etc. in case your mail HD fails.
#
echo "   Start du: ‘date‘" >> /var/log/sendlogs.status
du / 2> /dev/null | bzip2 -9 > /etc/info/logs/du.‘date +’%b%d%y’‘.bz2
# cp /etc/info/logs/du.‘date +’%b%d%y’‘.bz2 /to/some/other/HD
echo -e "DU: Created full file system du archive in /etc/info"
echo -e "----------------------------------------------------"


# Search for SUID programs, compare the results to the approved list and email
```

```
# the results
echo "   Start SUID: ‘date‘" >> /var/log/sendlogs.status
find / -type f \( -perm -04000 -o -perm -02000 \) -ls 2> /dev/null > /etc/info/suid-results-new
diff /etc/info/suid-results-checked /etc/info/suid-results-new 2> /dev/null > /etc/info/suid-results-diff
#
mail -s "$HOST SUID results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/suid-results-diff
rm -f /etc/info/suid-results-diff

echo -e "SUID: Sent SUID check.."
echo -e "----------------------"


# Search for rhost files, compare the results to the approved list and email
# the results
echo "   Start RHOSTs: ‘date‘" >> /var/log/sendlogs.status
find / 2> /dev/null | grep -e ".rhosts" -e "hosts.equiv" > /etc/info/rcmd-results-new
diff /etc/info/rcmd-results-checked /etc/info/rcmd-results-new > /etc/info/rcmd-results-diff
#
mail -s "$HOST RCMD results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/rcmd-results-diff
rm -f /etc/info/rcmd-results-diff

echo -e "Sent RCMD check.."
echo -e "-----------------"


# Search for altered RPM packages, compare the results to the approved list
# and email the results
echo "   Start RPMS: ‘date‘" >> /var/log/sendlogs.status
/bin/rpm -Va > /etc/info/rpm-results-new
diff /etc/info/rpm-results-checked /etc/info/rpm-results-new > /etc/info/rpm-results-diff
#
mail -s "$HOST RPM results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/rpm-results-diff
rm -f /etc/info/rpm-results-diff

echo -e "Sent RPM check.."
echo -e "----------------"


#Get SETI statsistics
#
# This section is commented out by default
#
#  (this is optional and only is useful for people using Seti and the Jsetidoor
#  proxy
#
#JDATE=‘cat /usr/src/archive/seti/proxy/jsetidoor/jseti-current-date‘
#JPERF="/usr/src/archive/seti/proxy/jsetidoor/jsd-performance.log"
#JLOG="/usr/src/archive/seti/proxy/jsetidoor/jsd.log"
#JCOUNT=‘cat $JLOG | grep -e $JDATE | grep -e update | wc --lines‘
#echo -e "\nSETI stats:  WU completed for $JDATE is $JCOUNT\n"
```

```
#echo -e "SETI stats:  WU completed for $JDATE is $JCOUNT" >> $JPERF
#
#Update date for next run
#/usr/src/archive/seti/proxy/jsetidoor/jseti-date


# This section is commented out by default
#
# This section is to DD one HD to a backup HD.  This is a simple but VERY
# effective online backup though it is only done once a night.  If you
# have a spare HD in your system, this is the next best thing to setting
# up RAID1.  Personally, I just recommend to setup RAID1!  :)
#
# Please note that the block size and timing was found by doing testing
#   for my specific system.  You should do this for your own setup to
#   to find your optimial setup.
#
#echo -e "-----------------------------------------------------------------------------"
#echo "   Start dd: `date`" >> /var/log/sendlogs.status
#echo -e "DD /dev/sda to /dev/sdc : 1k transfers yields an optimal 22minute"
#echo -e "transfer at 27 percent CPU load\n"
#time dd if=/dev/sda of=/dev/sdc bs=1k

echo -e "-----------------------------------------------------------------------------"
echo -e "\nRemaining entries are due to errors in the cron files or in /etc/logrotate.d files\n"


echo "Finished Sendlogs: `date`" >> /var/log/sendlogs.status
STOP=`date +%s`
echo -e "\n\nSendlogs took `echo "( $STOP - $START ) / 60" | bc -l` minutes\n"
```

```
#!/bin/sh

# TrinityOS-sendlogs.sh
# v01/07/01
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates:
#
# 01/07/01 - This script is now parsed directly from the SGML code and
#            because of this, several formatting issues were fixed.
#          - Made the output a little more pretty
#          - #ed out some diagnostic file information
#          - added an lsof log entry
```

```
#          - cleaned up the error reports in the SUID and RCMD searches
#
# 12/26/00 - Added --MARK-- Filtering
#
# 10/28/00 - Added an optional and #ed out section on DDing one HD to
#            another.  This is a simple but VERY effective online backup
#            though it is only done once a night.  If you have a spare HD
#            in your system, this is the next best thing to setting up
#            RAID1.  Personally, I just recommend to setup RAID1!  :)
#
# 10/08/00 - Deleted the removal of the SUID and RCMD new result files
#
# 09/16/00 - Added a full RPM database verification setup
#
# 04/15/00 - Added the $HOST variable to easily tune the SUBJECT field to
#            reflect the name of your Linux system.  You should edit this
#            to reflect your system.
#
# 04/09/00 - Hmmm.. we need %e and NOT %d for catching dates 01-09.
#            Basically, I need to reverve the change on 01/17/00.
#
# 02/21/00 - Doh!  We do need the spaces between %b and %d
#
# 01/17/00 - Fixed all the "date" issues.  Date now uses %d over %e and
#            doesn't use any spaces.
#
# 01/01/00 - Fixed a missing ">" on line 139
#
# 12/16/99 - Fixed the RCMD mailer command at the end.  The "mail -s" line
#            needed to be ONE line
#
# 11/26/99 - Cleaned things up a bit
#          - Made all file references absolute
#
# 02/01/99 - Added "w" to the vitals output


# Change this variable to reflect the HOSTNAME of this box
# ------------------------------------------------------
HOST="TrinityOS"


#Make sure that the "yesterdays-date" file exists.  If not, create it.
#
if [ -f /var/log/todays-date ]; then
     mv /var/log/todays-date /var/log/yesterdays-date;
  else
     date +'%b %e' > /var/log/yesterdays-date;
fi
```

```
#Make sure that the "/etc/info/logs" directory exists.  If not, create it.
#
if [ -a /etc/info ]; then
  if [ -a /etc/info/logs ]; then
       echo "";
    else
       mkdir /etc/info/logs;
  fi
  else
     mkdir /etc/info;
     mkdir /etc/info/logs;
fi


date +’%b %e’ > /var/log/todays-date

cat /var/log/messages | grep "‘cat /var/log/yesterdays-date‘" > /var/log/messlog.‘date +’%b%d%y’‘
export f1=/var/log/messlog.‘date +’%b%d%y’‘
export f2=/var/log/testfile
#echo "File 1: $f1"
#echo "File 2: $f2"

#For messages - FTP and PPP stuff
#
sed -e "/PWD/d" -e "/PASV/d" -e "/TYPE/d" -e "/PORT/d" -e "/NLST/d" -e "/SYST/d" $f1 > $f1.tmp
sed -e "/PASS/d" -e "/QUIT/d" -e "/LIST/d" -e "/CDUP/d" -e "/ATDT/d" -e "/Welcome/d" $f1.tmp > $f2.tmp
sed -e "/Using/d" -e "/Connect/d" -e "/Remote/d" -e "/IP address/d" -e "/CHECKSUM/d"  $f2.tmp > $f1.tmp
sed -e "/Terminated/d" -e "/Terminating/d" -e "/diald/d" -e "/2.2.0/d" -e "/Exit./d" $f1.tmp > $f2.tmp
sed -e "/(passwd=guest)/d" -e "/alarm/d" -e "/Failed/d" $f2.tmp > $f1.tmp

#For messages - modem specific stuff
#
#sed -e "/send /d" -e "/expect/d" -e "/OK/d" -e "/AT&F/d" -e "/ATZ/d" -e "/ ^M /d" $f1.tmp > $f2.tmp
#sed -e "/Swansea/d" -e "/logging/d" -e "/starting/d" -e "/Ready/d" -e "/0x03f8/d" -e "/0x02f8/d" $f2.tmp > $f1.t
#sed -e "/sbpcd.c/d" -e "/CR-563/d" -e "/copyright/d" -e "/sockets/d" -e "/Serial/d" -e "/registered/d" $f1.tmp >
#sed -e "/SLIP/d" -e "/sbpcd-0/d" -e "/ATMOX7/d" -e "/1.44M/d" -e "/8272A/d" -e "/statistics/d" $f2.tmp > $f1.tmp
#sed -e "/Please/d" -e "/hangup/d" -e "/ip-down/d" -e "/scans/d" $1.tmp -e "/abort on/d" $f1.tmp > $f2.tmp
#sed -e "/CONNECT /d" -e "/BUSY/d" -e "/SIGHUP/d" $f2.tmp > $f1.tmp

#For messages - modem dialout specific stuff
#
#echo -e "---------------------------------------" > /var/log/header.tmp
#echo -e "$HOST Call stats for \c" >> /var/log/header.tmp
#date >> /var/log/header.tmp
#echo -e "                                       " >> /var/log/header.tmp
#echo -e "Total number of connects: \c" >> /var/log/header.tmp
#grep -c "CONNECT" $f1.tmp >> /var/log/header.tmp
#echo -e "      21600: \c" >> /var/log/header.tmp
#grep -c "21600" $f1.tmp >> /var/log/header.tmp
```

```
#echo -e "      26400: \c" >> /var/log/header.tmp
#grep -c "26400" $f1.tmp >> /var/log/header.tmp
#echo -e "      28800: \c" >> /var/log/header.tmp
#grep -c "28800" $f1.tmp >> /var/log/header.tmp
#echo -e "      31200: \c" >> /var/log/header.tmp
#grep -c "31200" $f1.tmp >> /var/log/header.tmp
#echo -e "      33600: \c" >> /var/log/header.tmp
#grep -c "33600" $f1.tmp >> /var/log/header.tmp
#echo -e "      33600: \c" >> /var/log/header.tmp
#grep -c "41333" $f1.tmp >> /var/log/header.tmp
#echo -e "      41333: \c" >> /var/log/header.tmp
#grep -c "42666" $f1.tmp >> /var/log/header.tmp
#echo -e "      42666: \c" >> /var/log/header.tmp
#echo -e "                                        " >> /var/log/header.tmp
#echo -e "Total number of busys: \c" >> /var/log/header.tmp
#grep -c "BUSY" $f1.tmp >> /var/log/header.tmp
#echo -e "----------------------------------------" >> /var/log/header.tmp
#echo -e "                                        " >> /var/log/header.tmp
#cat /var/log/header.tmp >> $f1.tmp

#For messages - named specific stuff
#
sed -e "/Cleaned/d" -e "/USAGE/d" -e "/NSTATS/d" -e "/XSTATS/d" $f1.tmp > $f2.tmp
sed -e "/points/d" -e "/Lame server/d" $f2.tmp > $f1.tmp

#For messges - SSH specific
sed -e "/Generating /d" -e "/generation /d" -e "/NSTATS/d" -e "/XSTATS/d" $f1.tmp > $f2.tmp

#For messges - Delete --MARK-- entries
sed -e "/-- MARK --/d" $f2.tmp > $f1.tmp

mv $f1.tmp $f1
rm -R /var/log/*.tmp

mail -s "$HOST messages for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/messlog.‘date +’%b%d%y’‘

rm /var/log/messlog.‘date +’%b%d%y’‘

echo -e "-------------------------------------------------------"
echo -e "MESSAGES: Parsed, filtered, mailed and deleted messages"
echo -e "-------------------------------------------------------"

#----------------------------------------------

cat /var/log/syslog | grep "‘cat /var/log/yesterdays-date‘" > /var/log/syslog.‘date +’%b%d%y’‘

export f1=/var/log/syslog.‘date +’%b%d%y’‘
#echo "file 1: $f1"
#echo "file 2: $f2"
```

```
#Syslog - modem specific
#sed -e "/ got /d" -e "/abort on/d" -e "/expect/d" -e "/ ^M /d" -e "/AT&F1^M^M/d" $f1 > $f1.tmp
#sed -e "/ATZ^M^M/d" -e "/ATM0X7S11=40^M^M/d" -e "/Executed/d" -e "/ATDT/d" $f1.tmp > $f2.tmp
#sed -e "/Welcome/d" -e "/Using/d" -e "/Connect/d" -e "/Remote/d" -e "/IP address/d" $f2.tmp > $f1.tmp
#sed -e "/CHECKSUM/d" -e "/Terminated/d" -e "/Terminating/d" -e "/diald/d" -e "/2.2.0/d" $f1.tmp > $f2.tmp
#sed -e "/Exit./d" -e "/(passwd=guest)/d" -e "/alarm/d" -e "/Failed/d" -e "/CONNECT/d" $f2.tmp > $f1.tmp
#sed -e "/hangup/d" -e "/RINGING^M/d" $f1.tmp > $f2.tmp
#mv $f2.tmp $f1

#syslog FTP,
sed -e "/PWD/d" -e "/PASV/d" -e "/LIST/d" -e "/CDUP/d" -e "/RETR/d" -e "/CWD/d" $f1 > $f1.tmp
sed -e "/TYPE/d" -e "/PASS/d" -e "/QUIT/d" $f1.tmp > $f2.tmp

#For messages
sed -e "/send /d" -e "/expect/d" -e "/OK/d" -e "/AT&F/d" -e "/ATZ/d" -e "/ ^M /d" $f2.tmp > $f1.tmp
sed -e "/Swansea/d" -e "/logging/d" -e "/starting/d" -e "/Ready/d" -e "/0x03f8/d" $f1.tmp > $f2.tmp
sed -e "/0x02f8/d" -e "/sbpcd.c/d" -e "/CR-563/d" -e "/copyright/d" -e "/sockets/d" $f2.tmp > $f1.tmp
sed -e "/SLIP/d" -e "/sbpcd-0/d" -e "/1.44M/d" -e "/8272A/d" -e "/statistics/d" $f1.tmp > $f2.tmp
sed -e "/Please/d" -e "/hangup/d" -e "/ip-down/d" -e "/scans/d" $f2.tmp > $f1.tmp
sed -e "/abort on/d" -e "/Serial/d" -e "/registered/d" $f1.tmp > $f2.tmp

mv $f2.tmp $f1
rm -r /var/log/*.tmp 2> /dev/null > /dev/null

mail -s "$HOST syslog for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/syslog.‘date +’%b%d%y’‘
rm /var/log/syslog.‘date +’%b%d%y’‘

echo -e "SYSLOG: Parsed, filtered, mailed and deleted syslog"
echo -e "---------------------------------------------------"


cat /var/log/secure | grep "‘cat /var/log/yesterdays-date‘" > /var/log/secure.‘date +’%b%d%y’‘

export f1=/var/log/secure.‘date +’%b%d%y’‘
#echo "file 1: $f1"
#echo "file 2: $f2"

sed -e "/127/d" $f1 > $f1.tmp
mv $f1.tmp /var/log/secure.‘date +’%b%d%y’‘
mail -s "$HOST secure for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/secure.‘date +’%b%d%y’‘
rm -r /var/log/*.tmp
rm /var/log/secure.‘date +’%b%d%y’‘

echo -e "SECURE: Parsed, filtered, mailed and deleted secure"
echo -e "---------------------------------------------------"


cat /var/log/xferlog | grep "‘cat /var/log/yesterdays-date‘" > /var/log/xferlog.‘date +’%b%d%y’‘

mail -s "$HOST xferlog for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/xferlog.‘date +’%b%d%y’‘
```

```
rm /var/log/xferlog.‘date +’%b%d%y’‘

echo -e "XFERLOG: Parsed, filtered, mailed and deleted xferlog"
echo -e "-----------------------------------------------------"


cat /var/log/kernel | grep "‘cat /var/log/yesterdays-date‘" > /var/log/kernel.‘date +’%b%d%y’‘

mail -s "$HOST kernel for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/kernel.‘date +’%b%d%y’‘
rm /var/log/kernel.‘date +’%b%d%y’‘

echo -e "KERNEL: Parsed, filtered, mailed and deleted kernel"
echo -e "---------------------------------------------------"


df > /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
w >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
free >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
ps aux >> /var/log/sendlogs.‘date +’%b%d%y’‘
echo -e "\n\n\n" >> /var/log/sendlogs.‘date +’%b%d%y’‘
lsof -i >> /var/log/sendlogs.‘date +’%b%d%y’‘

mail -s "$HOST vitals for ‘cat /var/log/yesterdays-date‘" root@localhost < /var/log/sendlogs.‘date +’%b%d%y’‘
rm -f /var/log/sendlogs.‘date +’%b%d%y’‘

echo -e "VITALS: Sent system vitals.."
echo -e "----------------------------"


# Create a full file system ls-laR archive in /etc/info
#
#  NOTE:  You should ALSO copy this file to somewhere on a DIFFERENT HD,
#  floppy, etc. in case your mail HD fails.
#
ls -laR / 2> /dev/null | bzip2 > /etc/info/logs/ls-laR.‘date +’%b%d%y’‘.bz2
echo -e "LS-LAR: Created full file system ls-laR archive in /etc/info"
echo -e "------------------------------------------------------------"
# cp /etc/info/logs/ls-laR.‘date +’%b%d%y’‘.bz2 /to/some/other/HD


# Create a full file system du archive in /etc/info
#
#  NOTE:  You should ALSO copy this file to somewhere on a DIFFERENT HD,
#  floppy, etc. in case your mail HD fails.
#
du / 2> /dev/null | bzip2 > /etc/info/logs/du.‘date +’%b%d%y’‘.bz2
# cp /etc/info/logs/du.‘date +’%b%d%y’‘.bz2 /to/some/other/HD
```

```
echo -e "DU: Created full file system du archive in /etc/info"
echo -e "----------------------------------------------------"


# Search for SUID programs, compare the results to the approved list and email
# the results
find / -type f \( -perm -04000 -o -perm -02000 \) -ls 2> /dev/null > /etc/info/suid-results-new
diff /etc/info/suid-results-checked /etc/info/suid-results-new 2> /dev/null > /etc/info/suid-results-diff
#
mail -s "$HOST SUID results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/suid-results-diff
rm -f /etc/info/suid-results-new

echo -e "SUID: Sent SUID check.."
echo -e "-----------------------"


# Search for rhost files, compare the results to the approved list and email
# the results
find / 2> /dev/null | grep -e ".rhosts" -e "hosts.equiv" > /etc/info/rcmd-results-new
diff /etc/info/rcmd-results-checked /etc/info/rcmd-results-new > /etc/info/rcmd-results-diff
#
mail -s "$HOST RCMD results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/rcmd-results-diff
rm -f /etc/info/rcmd-results-new

echo -e "Sent RCMD check.."
echo -e "-----------------"


# Search for altered RPM packages, compare the results to the approved list
# and email the results
/bin/rpm -Va > /etc/info/rpm-results-new
diff /etc/info/rpm-results-checked /etc/info/rpm-results-new > /etc/info/rpm-results-diff
#
mail -s "$HOST RPM results for ‘cat /var/log/yesterdays-date‘" root@localhost < /etc/info/rpm-results-diff
rm -f /etc/info/rpm-results-diff

echo -e "Sent RPM check.."
echo -e "----------------"


# This section is commented out by default
#
# This section is to DD one HD to a backup HD.  This is a simple but VERY
# effective online backup though it is only done once a night.  If you
# have a spare HD in your system, this is the next best thing to setting
# up RAID1.  Personally, I just recommend to setup RAID1!  :)
#
# Please note that the block size and timing was found by doing testing
#   for my specific system.  You should do this for your own setup to
#   to find your optimial setup.
```

```
#
#echo -e "DD /dev/sda to /dev/sdd : 1k transfers yeilds an optimal 22minute transfer\n"
#time dd if=/dev/sda of=/dev/sdd bs=1k


echo -e "-------------------------------------------------------------------------------"
echo -e "\nRemaining entries are due to errors in the cron files or in /etc/logrotate.d files\n"
```

<Sendlogs STOP>

- Next, make the file executable by running "chmod 700 /usr/local/sbin/sendlogs"

- Now create the following directories and fix their permissions

```
                mkdir /etc/info
                mkdir /etc/info/logs
                chmod -R 700 /etc/info
```

* Before you run the "sendlogs" script, follow the procedure in 18 (Section 18)

- Now, you have to make cron run this script every day:

BSD-style (Slackware, etc): ———————————

Edit the file /var/spool/cron/crontabs/root and append the following:

```
                        --
                        # Run the sendlogs program at 12:00am everyday
                        0 12 * * * /usr/local/sbin/sendlogs
                        --
```

- That’s it. Now, make cron re-read it’s config files by doing:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}“

SysV-style (Redhat): ———————

Create the file /etc/cron.daily/a-sendlogs and enter in:

NOTE: Why the name "a-sendlogs"? The reason is because the crontab runs all the files in /etc/cron.daily in alphabetical order. We need to run the sendlogs script BEFORE the "rotatelogs" script executes.

```
                        #!/bin/sh
                        cd /usr/local/sbin
                        ./sendlogs
```

Now make it executable via "chmod 700 /etc/cron.daily/a-sendlogs"

9.7.1 Creating an off-line firewall hit log

Once you start getting the parsed nightly logs, I HIGHLY recommend that you start creating a on-going log file of your firewall hits. You can learn how to read the firewall hits in 10 (Section 10).

I do this by manually creating a simple ASCII text file that I populate with the date, port #, port type, the source name (manually found via nslookup), and the IP address. For the sites that won't reverse resolve, I just do a traceroute to the closest named hop.

So why do I do this? Because you'll soon see trends of simple telnets to full blown port scans from specific IPs and/or domains. Also.. some hackers run port scans that take weeks and not minutes. If you run a log like this, you'll catch them!

Here is one example from my "Firewall hits list" of some dirtbag that tried to do a DoS attack against my IMAP service. Not only did my firewall stop him, but TCP wrappers would have stopped him and I logged the fact. I've changed the IP address to protect the luser and myself.

NOTE: Not only is it important to log the destination port the hacker was trying to get to but also their source port. This luser was using source port 0 which is common DoS attack method:

```
        01/08/99       143/tcp Name:    cc6666666-b..nj.home.com       Address:  10.0.0.1
                from port 0!
```

9.7.2 Thoughts on various log entries you will see and what to do

Once you start seeing the proactive logs via email, some entries will seem bad at first but hopefully this section will help you understand what things mean:

- Proc Entries: The /proc file system is a virtual file system and somethings cannot be listed due to operating system restrictions and/or security issues. If you see entries like:

```
          ls: /proc/2/exe: No such file or directory
          ls: /proc/3/exe: No such file or directory
          ls: /proc/4/exe: No such file or directory
          ls: /proc/5/exe: No such file or directory
```

  Don't worry about it.. This is normal.

- Unexpected SUID file Changes: As part of keeping a system secure, you will need to patch it often. When you apply a new set of patches, the file size, date, etc. will change. The next Sendlogs results will notify you of these changes. If the changed files were due to an applied patch, things are ok.

  It should also be noted that as a Linux system is running, the EXT2 file system will eventually change a file's time stamp (typically after six months) from the file's creation DATE (month and day) and TIME (hour and minute) to simple the DATE (month, day, and year). So, when you see a file change from the Sendlogs script, definately make sure the file size and permissions are the same but pay close attention to the DATE. If only the date changed from the TIME to YEAR, things are ok.

- RPM database changes As you patch your system, you want to be sure that the changed files, RPM database, and the MD5 sums of files are accounted for. One nice thing about the RPM verification is that you can monitor if files are modified either on purpose, by corruption, or by intrusion.

So, part of maintaining a secure and reliable Linux box is you will have to replace the reference files in /etc/info. Once you are sure that the changes that have shown up in your email box are ok (as described above), you will need to move the new files to become the new reference file.

- SUID changes - Will have to be updated often since new patches will age

```
mv /etc/info/suid-results-new /etc/info/suid-results-checked
```

- RCMD changes - Won't need to be updated often

```
mv /etc/info/rcmd-results-new /etc/info/rcmd-results-checked
```

- RPM Changes - Will have to be updated often due to patches and/or corruption

```
mv /etc/info/rpm-results-new /etc/info/rpm-results-checked
```

10 Advanced firewall rule sets including IP Masquerade for single and multi-NIC setups

10.1 What is packet firewall

If you are unfamiliar with how TCP/IP packet filters work, the following should give you a decent start. Please understand that if you don't understand what is being described below, you should probably do a little research on how TCP/IP works.

Think of a IPCHAINS or IPFWADM rule set like the following:

- All interfaces (any network cards, PPP connections, the localhost interface, etc) on a Linux box have INPUT, OUTPUT, and FORWARD rules.
- What is the difference between DENY and REJECT? DENY:

  If you TELNET to a box that "denies" TELNET traffic, your TELNET client will just sit there and try for a period to connect to that remote host. Ultimately, the TELNET request will eventually timeout.

  REJECT:

  If you TELNET to a box that "rejects" TELNET traffic, your TELNET request traffic will be met with an ICMP message telling the originator that the traffic was rejected. This is the normal behavior for a machine that does not SUPPORT telnet server access in the first place (like stock versions of MS Windows9x, NT, etc.).

  - Why do I prefer REJECT over DENY? If someone connects to your server and you REJECT their traffic, it seems to them as if your computer cannot serve, say, TELNET connections. If you DENY the traffic, then their TELNET traffic just dies and their TELNET client eventually times out.

    So? With REJECT, a hacker doesn't know if your machine CAN or CAN NOT run a TELNET server. With DENY, a hacker will always KNOW that you are filtering them. I feel that a firewall using REJECTs make your box look "simpler" and thus less interesting to attack.

10.2 How a packet firewall works

So , lets explain how a packet firewall works with an example:

Say you have a TELNET packet (port 23) from the Internet that wants to reach your Linux box

1. The TELNET packet is sent from the remote computer on the Internet
2. The packet is received on PORT 23 to the INPUT rule on the -External NIC card-
3. If the TELNET packet is matched on the INPUT to allow the packet through:

   FYI: Some ideas of possible packet firewall rules can include:

   - source and destination IP addresses
   - TCP or UDP traffic
   - specific source and destination ports (TELNET, etc)
   - etc.

   Then let the packet IN though the packet firewall. If not matched, the packet is either REJECTED or DENIED. You can also log the fact that this packet was killed.
4. If passed, the TELNET packet then goes to the TELNET daemon on the Linux box to be processed. Once the reply TELNET traffic is generated, the actual return traffic will be returned on a HIGH PORT (port > 1024) and NOT on port 23.

   If you don't understand this, please read up on TCP/IP fundamentals since this discussion is out of the scope of TrinityOS.

   For this example, lets say the return TELNET traffic is on port 3200. Now, this return port 3200 traffic is then sent to the OUTPUT filter of the EXTERNAL NIC card.
5. If the packet is matched to allow the packet OUT, then let through. (like #3 above). If not matched, its either REJECTED or DENIED. You can also log the fact that this packet was killed.
6. Next, if the packet is on a DIFFERENT network than the destination address, the packet needs to be "forwarded". If the rule matches, forward the packet onto the correct network. If not matched, its either REJECTED or DENIED. You can also log the fact that this packet was killed. NOTE: This is is what a "router" does on a basic level.
7. If finally passed, the HIGH PORT packet leaves the Linux box to go over the Internet connection destined to that remote computer.

```
                                 +--------------------------------+
                                 |      Linux TCP/IP stack        |
                                 |________________________________|
                                 |     (3) Telnetd Server         |
                       {PORT 23} |________________________________| (Port 3200)
                       (2)  +--->| Input:    Forward:   Output:   |-------------+ (4)
                            |    +--------------------------------+             |
                            |                                                   |
                            |                                                   |
           +------------+   |                                  +------------+   |
           | Input      |   |                                  | Output     |<--+
           |  Rule      |   |                                  |  Rule      |   ^
{PORT 23}  |            |   |                                  |            |   |
```

```
  (1)  +-IN--->|  P a s s ? |---+                  +--------------|  P a s s ? |   |
       |       |     or     |                      |              |     or     |   |
       ^       |Deny/Reject?|                      |      (5)     |Deny/Reject?|   |
    ---------  +------------+                      |              +------+-----+   |
     *Send*          |                             |                     |         |
    ---------        v                      Check if packet              v         |
    Remote       Dump Packet       No +---- needs to be            Dump Packet     |
    Internet   (possibly log it)      |     forwarded           (possibly log it)  |
    site                              |            |                               |
    ---------                         |    (6)     | Yes                           |
    *Received*                        |            |                               |
    ---------                         |            v                               |
        ^                             |    +--------------+        +---------------^------+
        |         {PORT 3200}         |    | Forward      |        | Write the packet for |
    (7) +-----------------------------+    |   Rule       |        |  the destination     |
                                           |              |        |  network address     |
                                           |              |        |                      |
                        Dump Packet <------|Don't Forward?|        | Possibly re-write the|
                     (possibly log it)     |              |        |SRC addresses for MASQ|
                                           |   Forward?   |        +----------------------+
                                           |      or      |                       ^
                                           |FWD & MASQ it |-----------------------+
                                           +--------------+
```

10.3 How IP Masquerade (IP MASQ) works:

Basically, IP MASQ's main mechanism works when an INTERNAL machine initiates traffic to the outside world. External machines on the Internet **CAN** directly communicate to an internal machine(s) with the aid of PORTFWing but this is better explained in the IP Masquerade HOWTO. PORTFW support IS included in the TrinityOS firewall ruleset but for a full explination, again, please see the IP Masqerade HOWTO.

Anyway, when an internal machine (for now, in that diagram in the URL above, think of the "Remote Internet Site" on the left with your internal machine. If this diagram confuses you, just skip it and read through this example..

```
1. Say the internal machine trys to TELNET to some server out on the Internet.
   For this explict example, this example is:

        Source          src IP:    192.160.0.10
                        src port:  3200
                        dst port:  23

        Linux :         src IP:    111.222.212.222
        External        src port:  64000
                        dst port:  23

        Destination:    dest IP:   222.020.222.111
                        dst port:  23

2. The MASQ server receives this request from the MASQed PC over the Internal
```

interface and it hits the Input firewall. Here, the input firewall can either accept the packet or deny it. For this example, assume it will be ACCEPTed.

3. Now, if the packet was also allowed through the OUTPUT firewall, the TELNET would be finally forwarded through the MASQ server unchanged except...

3M. Notice that src port IP address of the TELNET is a private RFC1918 address? These addresses aren't routable on the Internet so it must be changed to a public address. To be able to track this change, the SRC port address will be changed as well.

The changes in IP address and port number is IP MASQ in action! What Masq basically does is RECORDs the traffic type (for this example, 23, TELNET), where the traffic is going (DST IP address, 222.020.222.111) and the original SRC port (SRC port 3200) from the MASQed client. It takes all this information and puts it into a MASQUERADE table.

It then will re-send this TELNET traffic out on its EXTERNAL NIC but it will also alter the packet. It will both re-addresses the Source IP address (SRC IP) with the MASQ server's own external IP address and change the source port (SRC port) to something in the range of 61000-64096. So, the packet would now look something like:

```
     Source:       SRC IP:   111.222.212.222
                   SRC port: 64000

     Destination:  DST IP:   222.020.222.111
                   DST port: 23
```

4. When the response comes back from that remote TELNET server, the Linux MASQ server will recognise that this traffic as coming back from a server that is in the MASQ table. It would then take the packet and first verify that it should be allowed through the INPUT section of the firewall. Next, it would then replace the destination IP address (DST IP) with the correct FINAL IP address of original internal TELNET client and also change the original SRC port address back to 3200.

The returning packet now looks like:

```
     Source:       DST IP:   222.020.222.111
                   DST port: 23

     Destination:  SRC IP:   192.160.0.10
                   SRC port  3200
```

```
Get it?


If you want another explination of how MASQ works, I wrote a semi-comprehensive
article about it in the August 1999 version of Linux Magazine.  You can get an
online version of it at:

        http://www.linux-mag.com/1999-08/guru_01.html
```

10.4 Differences between Packet and Statefull Firewalls

Now, I want to quickly comment on the use of HIGH TCP/IP ports and what is the difference between a PACKET firewall and a STATEFULLY INSPECTED firewall. Though you might let port 23 OUT of your Linux box (TELNET), if you don't also allow ports 1024-65535 back INTO your Linux box, TELNET won't work.

Now you might be thinking that letting in ALL high ports back into your Linux box is a BAD thing. You know what? YOU'RE RIGHT!

Realistically, it would be nice to only allow in only the return HIGH ports that you need. This is what the "-k" option in IPFWADM or "! -y" is for IPCHAINS. The problem is, IPFWADM and IPCHAINS aren't smart enough yet to understand all TCP/IP programs such like TELNET, WWW, SSH, etc. So, some programs you can lock down the high ports with the "-k" or "! -y" options while other programs will have to be configured to allow all 1024-65535 ports in.

Bummer huh? So your next question should be "Do others firewalls have this problem?" NO! Why? Because they use a technology called "Stateful Inspection".

Stateful firewalls actually listen to ALL network traffic step-by-step to make sure that everything is going 100% correctly.

Analogy:

Packet firewall: A packet firewall only checks for source and destination IP addresses and port numbers. Kinda like a strainer for different colored marbles (if one exists).

Stateful Firewall: A stateful firewall not only checks for source and destination IP addresses and port numbers, but it also LISTENS to all TCP/IP communications to make sure that all of the "communications" are following all procedures. Think of it as a realtime grammer and spell checker for "languages" like TELNET, WWW, etc. Hackers try to re-write the "language" to try to break into it, crash it, etc. A stateful firewall will see a given TCP/IP connection running a "language" like TELNET doing weird stuff that it shouldn't be doing and then it simply drops that weird packet. Much better huh?

So your next question should be: "I want a statefully inspected firewall for Linux and NOT a packet firewall. Where do I get one?!?!"

Well.. it now exists in IPTABLES under the 2.4.x kernels. This is a huge step for for Linux. Unfortunately, if you also need to use IP Masquerading (NAT), the MASQ support for some protocols under the 2.4.x kernel isn't on par with the 2.2.x kernel set. If you don't use IPMASQ, then then IPTABLES is a great solution. It should also be noted that non-IPMASQ users can still use their IPCHAINS rulesets under 2.4.x kernels with the aid of the ipchains.o kernel module.

For now, TrinityOS only covers IPCHAINS and an older IPFWADM ruleset. A IPTABLES ruleset is under developement but is a slow project as it is an entire rewrite and will offer far more features.

10.5 Debugging / Monitoring your firewall with examples

Once you setup one of the firewalls shown below, you might have some problems getting running or your might be getting strange new messages on the console. What do these messages mean?

In the below rule sets, any lines that either DENY or REJECT any traffic also have a "-o" to LOG this firewall hit to the SYSLOG messages file found either in:

Redhat: /var/log Slackware: /var/adm

If you look at one of these firewall logs, you would see something like:

The kernel logs this information looking like:

```
IPCHAINS:
Packet log: input DENY eth0 PROTO=17 12.75.147.174:1633 100.200.0.212:23
  L=44 S=0x00 I=54054 F=0x0040 T=254

IPFWADM:
Feb 23 07:37:01 Roadrunner kernel: IP fw-in rej eth0 TCP 12.75.147.174:1633
   100.200.0.212:23 L=44 S=0x00 I=54054 F=0x0040 T=254
```

There is a LOT of information in this just one line. Let break out this example so refer back to the original firewall hit as you read this. Please note that this example is for IPFWADM though it is DIRECTLY readable for IPCHAINS users.

NOTE: To understand all the various port numbers, protocol numbers, etc., I recommend you to go to the TOP URL in 5 (Section 5) and get all of the various documents from the IANA and put them in /etc/iana.

```
- This firewall "hit" occurred on: "Feb 23 07:37:01"

- This hit was on the "RoadRunner" computer.

- This hit occurred on the "IP" or TCP/IP protocol

- This hit came IN to ("fw-in") the firewall
        * Other logs can say "fw-out" for OUT or "fw-fwd" for FORWARD

- This hit was then "rejECTED".
        * Other logs can say "deny" or "accept"

- This firewall hit was on the "eth0" interface (Internet link)

- This hit was a "TCP" packet

- This hit came from IP address "12.75.147.174" on return port "1633".

- This hit was addressed to "100.200.0.212" to port "23" or TELNET.
        * If you don't know that port 23 is for TELNET, look at your
                  /etc/services file to see what other ports are used for.

- This packet was "44" bytes long

- This packet did NOT have any "Type of Service" (TOS) set
```

```
                --Don't worry if you don't understand this; not required to know
                * divide this by 4 to get the Type of Service for ipchains users

        - This packet had the "IP ID" number of "18"
                --Don't worry if you don't understand this; not required to know

        - This packet had a 16bit fragment offset including any TCP/IP packet
          flags of "0x0000"
                --Don't worry if you don't understand this; not required to know
                * A value that started with "0x2..." or "0x3..." means the "More
                  Fragments" bit was set so more fragmented packet will be coming in
                  to complete this one BIG packet.
                * A value which started with "0x4..." or "0x5..." means that the
                  "Don't Fragment" bit is set.
                * Any other values is the Fragment offset (divided by 8) to be later
                  used to recombinw into the original LARGE packet

        - This packet had a TimeToLive (TTL) of 20.
                * Every hop over the Internet will subtract (1) from this number.  Usually,
                  packets will start with a number of (255) and if that number ever reaches
                  (0), it means that realistically the packet was lost and will be deleted.
```

So, with basic understanding now, lets get either your MASQing or NON-MASQing Network up!

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++                                                                              ++
++  NOTE:  TrinityOS covers both IPCHAINS and IPFWADM firewall rule sets.       ++
++         --------------------------------------------------------------       ++
++                                                                              ++
++         ** Please note that the IPCHAINS ruleset is VASTLY more secure and   ++
++         and powerful when compared to the IPFWADM ruleset.  Due to the      ++
++         power and maintinance of IPCHAINS compared to IPFWADM, I recommend   ++
++         that any user that MUST run a 2.0.x kernel, that they patch their    ++
++         kernel to support IPCHAINS and use this newer ruleset                ++
++                                                                              ++
++         In the future, I will be replacing ALL rule sets with a modular      ++
++         system so all Secured IPs will be configured via a seperate file     ++
++         This will let users update their main firewall rule sets to newer    ++
++         verions without ANY manual customization for their environment.      ++
++                                                                              ++
++         This new system is already designed but I need to finish it up.      ++
++                                                                              ++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

- First, you need to make sure you have either the "ipchains" or "ipfwadm" or firewall programs. To check, run the commmand "whereis ipfwadm" or "whereis ipchains". If its there, you're set. If not, download it from the URL in 5 (Section 5)

* VERY IMPORTANT:

- All users should try to implement the following firewall rule set FIRST! Once you are sure that your network setup is working properly, then you can go back and secure things up. Ok?

- Next, create the file /etc/rc.d/rc.firewall

Slackware Users: DELETE the module info in the following IPFWADM rule set and put it in the /etc/rc.d/rc.modules file instead

- NOTE: If you don't plan to use some of these modules, comment or un-comment the various lines (I've already commented out cuseeme, irc, quake, and vdolive).

Edit the following file to use the proper configuration below depending if you are running a 2.2.x+ kernel (IPCHAINS) or a <2.0.x kernel (IPFWADM).

10.6 Simple IPCHAINS / IPFWADM rule set for initial IPMASQ testing

All of TrinityOS's step-by-step instructions, files, and scripts are fully scripted out for an automatic installation at:

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

The simple (WEAK) firewall rule set for IPCHAINS or IPFWADM :

```
--
#!/bin/sh

# Simple firewall rule set for both IPCHAINS and IPFWADM
# v3.00

echo "Enabling IP MASQ, MASQ timeouts, MASQ modules and simple firewalling"

#Load the MASQ modules
        #BSDComp
        /sbin/modprobe bsd_comp
        #
        echo Loading MASQ modules
        #/sbin/modprobe ip_masq_cuseeme
        /sbin/modprobe ip_masq_ftp
        #/sbin/modprobe ip_masq_irc
        #/sbin/modprobe ip_masq_quake
        #/sbin/modprobe ip_masq_vdolive

        # NOTE:  Though Real Audio will work without this module, the data
        #               will be coming in TCP mode vs. UDP mode.  With this
        #               module, you can enable UDP mode and possibly clean up
        #               any "glitches" in the sound stream
        /sbin/modprobe ip_masq_raudio

# Finished with MASQ modules

#   Multicast is a powerful, yet seldom used aspect of TCP/IP for multimedia
#       data.  Though it isn't used much now (because most ISPs don't enable
#       multicast on their networks, it will be very common in a few more
#       years.  Check out www.mbone.com for more detail.
#
#       NOTE:  Adding this feature is OPTIONAL
```

```
#
echo "Adding multicast route.."
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0

echo "Enabling IP Masqurading.."
echo "1" > /proc/sys/net/ipv4/ip_forward

        #Note:  Redhat users can enable this also by turning the
        #         flag forward flag on in /etc/sysconfig/network
        #
        #               Change the forward line to
        #                       FORWARD_IPV4=true


#--------------------------------------------------------------------------
# NOTE:  The following simple IPFWADM and IPCHAINS rule set is purely to
#            *test* IP MASQ functionality.
#
#               Though this rule set will work for
#               ALL users, it WILL NOT give you any good protection from lusers
#               (security crackers, etc) out on the Internet.  Trust me, now that
#               you are using a UNIX box, you need all the protection you can get!
#               Once you can confirm that is MASQ working properly, I *HIGHLY*
#               recommend that you -delete- this simple rc.firewall script and
#               replace it with the strong IPCHAINS or IPFWADM rule sets shown
#               later in this section!
#---------------------------------------------------------------------

#2.2.x+ kernels with IPCHAINS ONLY
#
echo "  - Setting Policies: IN/OUT is ACCEPT; FWD is reject (poor security; great functionality)"
/sbin/ipchains -P input ACCEPT
/sbin/ipchains -P output ACCEPT
/sbin/ipchains -P forward REJECT

echo "  - Flushing any old rule sets"
/sbin/ipchains -F input
/sbin/ipchains -F output
/sbin/ipchains -F forward

# 2.0.x kernels and IPFWADM users ONLY
#
#echo "  - Setting Policies: IN/OUT is ACCEPT; FWD is reject (poor security; great functionality)"
#/sbin/ipfwadm -I -p accept
#/sbin/ipfwadm -O -p accpet
#/sbin/ipfwadm -F -p reject

#echo "  - Flushing any old rule sets"
#/sbin/ipfwadm -I -f
#/sbin/ipfwadm -O -f
```

```
#/sbin/ipfwadm -F -f

echo "Extending MASQ timeouts.."
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  60 sec timeout for UDP traffic (Important for MASQ'ed ICQ users)
#
# IPCHAINS
/sbin/ipchains -M -S 7200 10 60
#
# IPFWADM
#/sbin/ipfwadm -M -s 7200 10 60


echo "Enable IP Masq.."
#
#IPCHAINS
ipchains -A forward -s 192.168.0.0/24 -j MASQ
#
#IPFWADM
#/sbin/ipfwadm -F -a m -S 192.168.0.0/24 -D 0.0.0.0/0 -W eth0

echo "rc.firewall done."
----
```

Next, append this to the end of the "/etc/rc.d/rc.local" file

All distributions:

```
        --
        #Run the IP MASQ and firewall script
        /etc/rc.d/rc.firewall
        --
```

- Finally, make the rc.firewall file ROOT executable ONLY

```
        chmod 700 /etc/rc.d/rc.firewall
```

That's it. Go ahead and run the new ruleset by typing in **/etc/rc.d/rc.firewall** and make sure that the Linux box can still access the Internet both by IP address and DNS names. For Masquerade users, also make sure that INTERNAL masqed PCs can access the Internet by both methods. If things do NOT work for you, please see Section 5 of the IP Masquerade HOWTO at `<http://www.ecst.csuchico.edu/~dranch/LINUX/ipmasq/c-html/>`. This document will help you troubleshoot any issues.

Once you confirm that IP-MASQ works ok, it is *HIGHLY* recommended to replace the above WEAK rule sets with one of the below STRONG rule sets.

```
##############################################################################
# MASQ rc.firewall                                                           #
#                                                                            #
#  - There are -3- rule sets listed below:                                   #
#                                                                            #
```

```
#     1. Strong rc.firewall rule set for IPCHAINS w/ and w/o MASQ support   #
#        for single, dual, and even three NIC configurations.               #
#                                                                           #
#         ^^ This is current the ONLY rule set that is maintained ^^        #
#                                                                           #
#     2. Strong rc.firewall rule set for IPFWADM w/ MASQ support            #
#                                                                           #
#     3. Strong rc.firewall rule set for IPFWADM w/o MASQ support for       #
#        single NIC Linux boxes.                                            #
#                                                                           #
#  - As mentioned above, once you have confirmed that the initial MASQ      #
#    functionality, You *SHOULD* either create your own strong firewall     #
#    rule set or use the following TrinityOS firewall rule set.             #
#                                                                           #
#############################################################################
```

*** If you aren't running MASQ, check out the other firewall rule set that follows after this one. ***

NOTE: You will have to edit this to allow machines you care about into your machine. All of this is well commented though.

NOTE #2: Even if you aren't running MASQ, you should modify these rule sets to suit your needs and APPLY them!!! You DO need some protection from the Internet!

——————————————————————————

All of TrinityOS's step-by-step instructions, files, and scripts are fully scripted out for an automatic installation at:

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

or you can just get the file here: `<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS-security/etc/rc.d/>`

It is HIGHLY recommended that you get the rc.firewall and the other TrinityOS scripts from the TrinityOS-Security archive (URL above) as it will help avoid typos, etc. *** Do NOT try to cut and paste the various scripts via a web browser into a text editor. If you do this, you will most likely find that the resulting scripts will have formatting errors (thus syntax errors) and also most likely every line will have ^M characters at the end of it which will abnormally terminate the script trying to be run. ——————————————————————————

```
+-----------------------------------------------------------------+
| rc.firewall for MASQ setups with a STRONG IPCHAINS RULE SET for  |
|               2.4.x, 2.2.x, and patched 2.0.x. kernels          |
+-----------------------------------------------------------------+
```

CRITICAL NOTE:

- All kernel versions less than 2.2.20 have a symlink vunerability. Upgrade now.
- ALL kernel versions less than 2.2.16 have a TCP exploit that when combined with tools such as Sendmail, will lead to a root compromise.
- All kernels below 2.2.12 have a IP fragmentation bug. This will make ALL strong IPCHAINS rule sets vulnerable! Upgrade NOW!

10.7 Strong TrinityOS IPCHAINS firewall rule set

/etc/rc.d/rc.firewall

<TrinityOS rule set START>

```
#!/bin/sh

# ----------------------------------------------------------------------------
FWVER="v4.21-123nic"
#
# Part of the copyrighted and trademarked TrinityOS document.
# http://www.ecst.csuchico.edu/~dranch/LINUX/index-linux.html
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
#    You may use this file for private or internal commercial use ONLY.
#
#    Any duplication and/or use of this file or its contents for direct
#    commercial (commercial being for profit) applications and/or
#    written publications (be it for profit OR free) must be granted
#    by written permission from David Ranch.  Basically, just ASK me..
#    I'm a pretty easy going guy but DON'T assume anything.  Ok?
#
# Sorry for the harsh language here but the TrinityOS ruleset has been
#  taken advantage of recently.
#
# --
# Summary:
#
#    The TrinityOS ruleset is a comprehensive IPCHAINS ruleset that
#    supports filtering for 1, 2, and 3 network interfaces.  This allows
#    for strong filtering for simple one interface PPP users, two interface
#    MASQ users, and even three interface MASQ users with a DMZ segment.  In
#    addition to all this, TrinityOS allows to explictly filter various types of
#    traffic including ICMP, known trojan horse traffic, etc.
#
#    NOTE:  The current 4.00 firewall version requires that the INTIF
#           (internal) interface be configured to then allow for the INT2IF
#           (DMZ network) to function.  If there is enough requests, I can
#           rework the ruleset to let INTIF and INT2IF load independantly.
#
# ----------------------------------------------------------------------------
#   You can get this file at:
#
# http://www.ecst.csuchico.edu/~dranch/LINUX/index-linux.html#trinityos
# ----------------------------------------------------------------------------
#
# Personal Changes:
#
```

```
#   Put any of your own version notes HERE.  Its a good idea to document
#   what you've changed.
#
# -----------------------------------------------------------------------------
#
# TrinityOS Rule Set History:
#
# 04/16/05 - 4.21
#            - Updated the bogon list to reflect changed bogon listing and
#              added output Multicast and NFS traffic filters

# 01/29/03 - 4.20
#            - The INT2BROAD variable was missing for the DMZ configuration
#              but the proper setting was being automatically used regardless.
#
# 01/13/03 - 4.10
#            - The latter half of the OUTPUT section was using $UNIVERSE/0
#              instead of $UNIVERSE which was already set to 0.0.0.0/0.
#              This was a harmless typo and didn't hurt anything but was
#              incorrect
#
# 12/30/01 - 4.05
#            - Somehow ip_forward was getting set to "0" instead of "1"
#            - Added comments when a 2.4.x kernel is found that running
#              IPCHAINS emulation is NOT recommended due to poor MASQ
#              support.  It is recommended to run a native IPTABLES ruleset
#              under 2.4.x kernels.
#
# 12/01/01 - 4.03
#            - Added an echo statement to let things run if you dont use
#              DHCP
#            - Added filters for the SubSeven trojan
#            - Added comments to let peopel know that NOT having the
#              ip_dynaddr or ip_defrag option is ok
#
# 11/09/01 - 4.02
#          - Disabled external DNSd and SMTPd server options as per the
#            default.
#          - Added comments and #ed out DHCPd for eth1 (input and output)
#          - split up the SSHd and DNSd enable/disable area for eth1
#          - #ed out SSHd and DNSd access (output) per the correct default
#
# 10/04/01 - 4.01f
#          - added ipchains check for 2.4.x kernels
#          - make sure that dhcpc is really enabled by default
#          - Added a logger line to send final result to SYSLOG
#
# 09/06/01 - v4.01
#          - Fixed some syntax issues with left/right parens
#          - replaced all the bash -n if..thens with string checks since
```

```
#            it seems that bash doesnt know what to do with non-initialized
#            vars
#          - ** check for all foo entries
#
# 09/03/01 - v4.00
#
#          - Changed the DMZ section to now allow full SSH connectivity between
#            the DMZ and internal NICs.
#          - Moved the INPUT DMZ-specific ALLOW/REJECT section to be below the
#            input SECUREHOST section
#          - Updated and rearranged the debug logging section
#          - Added #ed out support for the H.323 IPMASQ module
#          - Added PPTP support for MASQed clients
#
# 06/20/01 - v3.85
#          - The IPCHAINS ruleset now can support single interface machines
#            for those users who just want a firewall but aren't MASQing, etc.
#          - To enable this new feature, the INTIF variable (internal interface)
#            needs to be set but left EMPTY.  With this set, the other INTIF
#            sections will be disabled via IF..THEN checks.
#
# 03/20/01 - v3.83d-3NIC
#
# - Added 3rd NIC (eth2) for DMZ applications like 802.11b wireless networks
#
#    eth0 = Internet                           [  public IP   ]
#    eth1 = internal trusted net               [ 192.168.0.x  ]
#    eth2 = DMZ wireless network (not trusted) [ 192.168.10.x ]
#
#          This DMZ interface can ONLY do the following globally
#                               - DHCP, DNS, internet WWW, internet FTP
#              - SSH (to the internet and devices on the INT interface
#                (eth1)
#              - ping machines on the Internet AND devices on eth1
#
#                      This interface CANNOT
#                               - accept FTP
#              - SSH any hosts on eth1
#
#   The reason that I implimented this DMZ setup is for wireless networks.
#   Ultimately, the 802.11b WEP encryption spec is flawed and can be completely
#   sniffed within a matter of hours.  Because of this, you should ONLY allow
#   encrypted streams:  SSH, IPSEC, and maybe PPTP.
#
# v3.83d - 03/06/01
#     - Fixed a typo (stray #) where the RFC1918 10.x.x.x network was
#       NOT being filtered in the OUTPUT section
#
# v3.83c - 01/27/01
#     - Fixed a wrong output netmask for NET-TEST-B being a /12 instead
```

```
#       of a /16.  But, this really doesn't matter as I have disabled
#       the filtering of reserved IP space as ARIN constantly is releasing
#       this address space to the public without any form of notification.
#       See the update for v3.83a
#
# v3.83b - 01/06/01
#     - Fixed a missing ".0" in the Reserved-7 filters for the 72.0.0
#       networks
#
# v3.83a - 11/09/00
#     - Deleted all non RFC1918 address filtering.  It seems that many of the
#       addresses that the IANA reports as "reserved" are actually in use.
#
#     - Removed all rc.firewall history motes from v3.60 and older to
#       the TrinityOS-old-updates.wri (URL is above)
#
# v3.82 - 10/28/00
#     - Updated the port range for Xwindows filtering
#
# v3.81 - 10/15/00
#     - Crap!  Last subnet error in the Reserved-8 IANA section.  Please
#       change the subnet mask on 68.0.0.0 to a /6!
#
# v3.80 - 10/13/00
#     - Updated the version since this really is a big update
#
# ------------------------------------------------------------------------------
# All changes older than version 3.80 have been moved to the archives available
#   at:
#
#       <"http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS-old-updates.wri">
#-------------------------------------------------------------------------------

#--------------------------------------------------------------------
# This configuration assumes the following (DSL / Cablemodem setup):
#
#       1) The external interface is running on "eth0"
#       2) The external IP address is dynamically or statically assigned
#       3) The optional internal interface is "eth1"
#       4) The internal network is addressed within the private
#           192.168.0.x TCP/IP addressing scheme per RFC1918A
#       5) The optional DMZ network is on eth2
#
#   ****
#   NOTE:  All 2.2.x Linux kernels prior to 2.2.16 have TCP exploit that
#   ****   that when combined with tools like Sendmail can leed to a ROOT
#          compromise.  In addition to this, all kernels less than 2.2.11 have
#          a fragmentation bug that renders all strong IPCHAINS rule sets void.
#          It is CRITICAL that users upgrade the Linux kernel to at least a
#          2.2.16+ kernel for proper firewall and system security.
```

```
#
#--------------------------------------------------------------------

#********************************************************************
# Initializing
#********************************************************************
echo -e "\n\nLoading TrinityOS IPCHAINS Firewall $FWVER"
echo "--------------------------------------------------------------------"

#--------------------------------------------------------------------
# Variables
#--------------------------------------------------------------------

# The loopback interface and address
#
LOOPBACKIF="lo"
LOOPBACKIP="127.0.0.1"

# External interface device.
#
# NOTE: PPP and SLIP users will want to replace this interface
#       with the correct modem interface such as "ppp0" or "sl0"
#
#     For users that might have multiple PPP interfaces, you can
#         try the following code.  You will need to call the firewall
#         from /etc/ppp/ip-up script with a "$1" appended at the end.
#
#if [ "x$1" != "x" ]; then
#   EXTIF=$1
#else
#   EXTIF="ippp0"
#fi
#
EXTIF="eth0"

# Make sure the external interface is up
if ! /sbin/ifconfig | grep $EXTIF > /dev/null; then
  echo -e "\n\nExternal interface is DOWN.  Aborting."
  exit 1;
fi
echo External Interface: $EXTIF

# IP address of the external interface
#
#   *
#   * If you get a DYNAMIC IP address (regardless if you use PPP
#   * with a modem or DHCP with Ethernet), you *MUST* make this firewall
#   * rule set understand your new IP address everytime you get a new
#   * IP address.  To do this, enable the following one-line script.
#   *
```

```
#
#   (Please note that the different single and double quote characters MATTER).
#
# NOTE: Red Hat v6.0 users who run DHCP to get TCP/IP addresses
#       (Cablemodems, DSL, etc) will need to install and use a different
#       DHCP client than the stock client called "pump".  Redhat 6.2+
#       comes with a newer version of "pump" that CAN run scripts upon
#       lease bringup, renew, etc. but older versions are broken.
#
#       The reason for this whole issue is the old "pump" doesn’t support the
#       ability to run scripts run when DHCP gets an IP address.
#       Specifically, DHCP doles out IP addresses to its clients for
#       limited amounts of time; this is called a "lease".
#       When a DHCP "lease" expires, the client will query the DHCP
#       server for a "lease renewal".  Though the DHCP client will
#       usually get back its original IP address in the renewal, this
#       is NOT always guaranteed.  With this understood, if your DHCP
#       client receives a different IP address than the IPCHAINS
#       firewall was configured for, the firewall will block ALL
#       network access in and out of the Linux server because that
#       is what it was configured to do.
#
#       As mentioned above, the key to solve this problem is to use a
#       DHCP client program, such like DHCPcd found in Section 5, that
#       can re-run the /etc/rc.d/rc.firewall rule set once a new TCP/IP
#       address is set.  The new rule set will then make the required
#       changes to the rule sets to allow network traffic from and to
#       your new TCP/IP address.
#
#       With the dhcpcd program, it will need to be executed with a
#       specific command line option to have the firewall rule set
#       re-run upon every DHCP lease renew (please note the -c syntax
#       is depreciated in newer DHCPcd clients).  Please see the
#       DHCPcd section in TrinityOS for full details on how to edit
#       the /sbin/ifup file.
#
#
# Static TCP/IP addressed users: For EXTIP, EXTBROAD, and EXTGW, simply replace
# the pipelines with your correct TCP/IP address, broadcast address, and
# external gateway, respectively.
#
# e.g.:   EXTIP="100.200.0.212"
#
EXTIP=‘/sbin/ifconfig $EXTIF | awk ’/inet addr/ { gsub(".*:", "", $2) ; print $2 }’‘

if [ "$EXTIP" = ’’ ]; then
   echo "Aborting: Unable to determine the IP of $EXTIF ... DHCP or PPP problem?"
   exit 1
fi
```

```
echo External IP: $EXTIP



# Broadcast address of the external network
#
# Static TCP/IP addressed users:
#
# Simply delete all of the text and including the single quotes and
# replace it with your correct TCP/IP netmask enclosed in double
# quotes.
#
# e.g.:   EXTBROAD="100.200.0.255"
#
EXTBROAD=`/sbin/ifconfig $EXTIF | awk '/inet addr/ { gsub(".*:", "", $3) ; print $3 }'`
echo External broadcast: $EXTBROAD

# Gateway for the external network
#
# Static TCP/IP addressed users:
#
# Simply delete all of the text and including the single quotes and
# replace it with your correct TCP/IP default gateway or "next hop
# address".
#
# e.g.:   DGW="100.200.0.1"
#
EXTGW=`/sbin/route -n | grep -A 4 UG | awk '{ print $2}'`
echo Default GW: $EXTGW

echo " --- "

# Internal interface device.
#
#  ** READ ME:
#
#    If you don't have any other interfaces than say eth0, delete the
#    word "eth1" below.  i.e. make it read:
#
#        INTIF=""
#
INTIF=""
if [ "$INTIF" != "" ]; then
    echo "Internal Interface: $INTIF"
  else
    echo -e "Internal Interface: None\n**  MASQ and DMZ support disabled**"
fi

if [ "$INTIF" != "" ]; then
  # IP address on the internal interface
```

```
  #
  #  ** READ ME:
  #
  #    If you don't have any other interfaces, delete the address
  #    "192.168.0.1" but leave the rest.  i.e.  INTIP=""
  #
  INTIP=""
  echo Internal IP: $INTIP
fi

if [ "$INTIF" != "" ]; then
  # IP network address of the internal network
  #
  #  ** READ ME:
  #
  #    If you don't have any other interfaces, delete the address
  #    "192.168.0.0/24" but leave the rest.  i.e.  INTLAN=""
  #
  INTLAN=""
      echo Internal LAN: $INTLAN
fi

echo " --- "


#Do not remove this check as the ruleset currently requires the INTIF
#interface to exist for the INT2IF interface to properly function.
#
if [ "$INTIF" != "" ]; then
  # DMZ interface device.
  #
  #  ** READ ME:
  #
  #    If you don't have any other interfaces than say eth0, delete the
  #    word "eth2" below.  i.e. make it read:
  #
  #        INT2IF=""
  #
  #INT2IF="eth2"
  INT2IF=""
  if [ "$INT2IF" != "" ]; then
      echo "DMZ network interface: $INT2IF"
    else
      echo -e "DMZ Interface: None\n  **DMZ support disabled**"
  fi

  if [ "$INT2IF" != "" ]; then
    # IP address on the DMZ interface
    #
    #    If you don't have any other interfaces, delete the address
```

```
    #    "192.168.10.1" but leave the rest.  i.e.  INT2IP=""
    #
    INT2IP=""
    echo "DMZ interface IP: $INT2IP"
  fi

  if [ "$INT2IF" != "" ]; then
    # IP network address of the DMZ network
    #
    #    If you don't have any other interfaces, delete the address
    #    "192.168.10.0/24" but leave the rest.  i.e.  INT2LAN=""
    #
    INT2LAN=""
        echo DMZ network subnet: $INT2LAN
  fi

  if [ "$INT2IF" != "" ]; then
    # IP network broadcast of the DMZ network
    #
    #    If you don't have any other interfaces, delete the address
    #    "192.168.10.255" but leave the rest.  i.e.  INT2BROAD=""
    #
    INT2BROAD=""
        echo DMZ network broadcast: $INT2BROAD
  fi
fi


echo " --- "


# IP Mask for all IP addresses
UNIVERSE="0.0.0.0/0"

# IP Mask for broadcast transmissions
BROADCAST="255.255.255.255"

# Specification of the high unprivileged IP ports.
UNPRIVPORTS="1024:65535"

# Specification of X Window System (TCP) ports.
XWINDOWS_PORTS="6000:6063"


# The TCP/IP addresses of a specifically allowed EXTERNAL hosts
#
#   NOTE:  If you want to allow in an ENTIRE NETWORK, let the
#          last octet of the network be a .0 and add the netmask.
#            e.g.:
#                        SECUREHOST="200.244.0.0/26"
```

```
#
# Disabled by default.
#
#SECUREHOST="200.211.0.40"
#echo Secure Host1 IP: $SECUREHOST
#SECUREHOST2="200.211.0.41"
#echo Secure Host2 IP: $SECUREHOST2
#SECUREHOST3="200.244.0.42"
#echo Secure Host3 IP: $SECUREHOST3
#SECUREHOST4="200.244.0.43"
#echo Secure Host4 IP: $SECUREHOST4
#SECUREHOST5="200.244.0.44"
#echo Secure Host4 IP: $SECUREHOST5


# The TCP/IP addresses of a specifically allowed DMZ hosts
#
#   NOTE:  If you want to allow in an ENTIRE NETWORK, let the
#          last octet of the network be a .0 and add the netmask.
#            e.g.:
#                       DMZHOST1="192.168.10.10"
#
# Disabled by default.
#
#DMZHOST1="192.168.10.10"
#echo DMZ Secure Host1 IP: $DMZHOST1
#DMZHOST2="192.168.10.20"
#echo DMZ Secure Host2 IP: $DMZHOST2


# IP Port Forwarded Addresses
#
# Port forwarding allows external traffic to directly connect to an INTERNAL
# Masq'ed machine. An example need for port forwarding is the need for external
# users to directly contact a WWW server behind the MASQ server.
#
# To enable portfw, you need to un-# out and edit the lines above for one or
# more SECUREHOSTs.  You then need to un-# out the PORTFW in the FORWARD
# sections of later in the rule set.
#
# If you want to simply portfw one explicit host, it should be configured via a
# SECUREHOST option above.  If this PORTFW'ed port should be available for ALL
# hosts on the Inet, it should be opened up in the INPUT section much like for
# HTTP, Sendmail, etc.
#
# NOTE: Port forwarding is well beyond the scope of this documentation to
#       explain the security issues implied in opening up access like this.
#       Please see Appendix A to find the IP-MASQ-HOWTO for a full explanation.
#
# Disabled by default.
```

```
#
#PORTFWIP1="192.168.0.20"
#echo PortFW1 IP: $PORTFWIP1
#PORTFWIP2="192.168.0.20"
#echo PortFW2 IP: $PORTFWIP2
#PORTFWIP3="192.168.0.20"
#echo PortFW3 IP: $PORTFWIP3


# TCP/IP addresses of INTENRAL hosts network allowed to directly
#       connect to the Linux server.  All internal hosts are allowed
#       per default.
#
# Disabled by default
#HOST1IP="192.168.0.10"
#echo Internal Host 1 IP: $HOST1IP
#HOST2IP="192.168.0.11"
#echo Internal Host 2 IP: $HOST2IP

# Logging state.
#
# Uncomment the " " line and comment the "-l" (please note is this a
# lower case "L" and NOT a numerial one) line if you want to
# disable logging of some of more important the IPCHAINS rule sets.
#
# The output of this logging can be found in the /var/log/messages
# file.  It is recommended that you leave this setting enabled.
# If you need to reduce some of the logging, edit the rule sets and
# delete the "$LOGGING" syntax from the rule set that you aren't
# interested in.
#
# LOGGING=" "
echo "Logging is: ENABLED"
LOGGING="-l"

echo " --- "

#Verify that IPCHAINS is loaded for 2.4.x kernels
#
if [ -n "`/bin/uname -a | awk {'print $3'} | grep 2.4`" ]; then
  echo "Running 2.4.x kernel"
  echo "  - Please note that running IPCHAINS emulation under a 2.4.x"
  echo "    is NOT recommended as various MASQ modules such as FTP, etc"
  echo "    will no longer function.  To regain this functionality, you"
  echo -e "    MUST run a native IPTABLES ruleset.\n"

  if [ -z "`/sbin/lsmod | grep ipchains`" ]; then
      echo "loading ipchains.o"
      /sbin/insmod ipchains
    else
```

```
     echo "  ipchains.o already loaded."
  fi
fi

echo " --- "

echo "----------------------------------------------------------------------"

#--------------------------------------------------------------------
# Debugging Section
#--------------------------------------------------------------------
# If you are having problems with the firewall, uncomment the lines
# below and then re-run the firewall to make sure that the firewall
# is not giving any errors, etc.  The output of this debugging
# script will be in a file called /tmp/rc.firewall.dump
#--------------------------------------------------------------------
#
#echo "  - Debugging."
#echo Loopback IP: $LOOPBACKIP > /tmp/rc.firewall.dump
#echo Loopback interface name: $LOOPBACKIF >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo External interface name: $EXTIF >> /tmp/rc.firewall.dump
#echo External interface IP: $EXTIP >> /tmp/rc.firewall.dump
#echo External interface broadcast IP: $EXTBROAD >> /tmp/rc.firewall.dump
#echo External interface default gateway: $EXTGW >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo Internal interface name: $INTIF >> /tmp/rc.firewall.dump
#echo Internal interface IP: $INTIP >> /tmp/rc.firewall.dump
#echo Internal LAN address: $INTLAN >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo DMZ interface name: $INT2IF >> /tmp/rc.firewall.dump
#echo DMZ interface IP: $INT2IP >> /tmp/rc.firewall.dump
#echo DMZ LAN address: $INT2LAN >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo External secured host: $SECUREHOST >> /tmp/rc.firewall.dump
#echo External secured host #2: $SECUREHOST2 >> /tmp/rc.firewall.dump
#echo External secured host #3: $SECUREHOST3 >> /tmp/rc.firewall.dump
#echo External secured host #4: $SECUREHOST4 >> /tmp/rc.firewall.dump
#echo External secured host #4: $SECUREHOST5 >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo DMZ secured host #1: $DMZHOST1 >> /tmp/rc.firewall.dump >> /tmp/rc.firewall.dump
#echo DMZ secured host #2: $DMZHOST2 >> /tmp/rc.firewall.dump >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump

#--------------------------------------------------------------------
# General
#--------------------------------------------------------------------
# Performs general processing such as setting the multicast route
# and DHCP address hacking.
#
```

```
# Multicast is a powerful, yet seldom used aspect of TCP/IP for multimedia
# data. Though it isn't used much now (because most ISPs don't enable multicast
# on their networks, it will be very common in a few more years. Check out
# www.mbone.com for more detail.
#
# Adding this feature is OPTIONAL.
#
# Disabled by default.
#echo "  - Adding multicast route."
#/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev $EXTIF


# Disable IP spoofing attacks.
#
# This drops traffic addressed for one network though it is being received on a
# different interface.
#
echo "  - Disabling IP Spoofing attacks."
for file in /proc/sys/net/ipv4/conf/*/rp_filter
do
 echo "2" > $file
done

# Comment the following out of you are not using a dynamic address
#
#  Please note that some kernels dont have this enabled.
#  If this option gives an error, you can safely ignore it.
#
echo "  - Enabling dynamic TCP/IP address hacking."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr


# Enable TCP SYN Cookie protection:
#
echo "  - Enable TCP SYN Cookie protection"
echo "1" > /proc/sys/net/ipv4/tcp_syncookies

# Ensure that various ICMP sanity settings are there
#
echo "  - Enable ICMP sanity settings"

# Disable ICMP broadcast echo protection
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Enable bad error message protection
echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

# Disable ICMP Re-directs
for file in /proc/sys/net/ipv4/conf/*/accept_redirects; do
  echo "0" > $file
```

```
done
#

# Ensure that source-routed packets are dropped
#   - If you are running IPROUTE2, this will need to be DISABLED
#
echo "  - Ensure that source-routed packets are dropped "
for file in /proc/sys/net/ipv4/conf/*/accept_source_route; do
   echo "0" > $file
done

# Log spoofed, source-routed, and redirect packets
#
echo "  - Log spoofed, source-routed, and redirect packets "
for file in /proc/sys/net/ipv4/conf/*/log_martians; do
  echo "1" > $file
done

#----------------------------------------------------------------------
# Type of Service (TOS) Settings
#----------------------------------------------------------------------
# Though very FEW ISPs do anything with the TOS bits, I thought you'd
# like to see it.  In theory, you can tell the Internet how to handle
# your traffic, be it sensitive to delay, throughput, etc.
#
#       -t 0x01 0x10 = Minimum Delay
#       -t 0x01 0x08 = Maximum Throughput
#       -t 0x01 0x04 = Maximum Reliability
#       -t 0x01 0x02 = Minimum Cost
#
# Example:
#
# Settings for FTP, SSH, and TELNET
# /sbin/ipchains -A output -p tcp -d 0/0 21:23  -t 0x01 0x10
#
# Settings for WWW
# /sbin/ipchains -A output -p tcp -d 0/0 80 -t 0x01 0x10


# Dont run these commands if MASQ isnt compiled into the kernel
if [ -a /proc/sys/net/ipv4/ip_always_defrag ] && [ "$INTIF" != "" ]; then

  #----------------------------------------------------------------------
  # Masquerading Timeouts
  #----------------------------------------------------------------------
  # Set timeout values for masq sessions (seconds).
  #
  # Item #1 - 2 hrs timeout for TCP session timeouts
  # Item #2 - 10 sec timeout for traffic after the TCP/IP "FIN" packet is received
  # Item #3 - 60 sec timeout for UDP traffic
```

```
  #
  # Note to ICQ users:  You might want to set the UDP timeout to something
  #                     like 160.
  #
  echo "  - Changing IP masquerading timeouts."
  /sbin/ipchains -M -S 7200 10 60
fi

# Dont run these commands if MASQ isnt compiled into the kernel
if [ -a /proc/sys/net/ipv4/ip_always_defrag ]; then

  #-------------------------------------------------------------------
  # Masq Modules
  #-------------------------------------------------------------------
  # Most TCP/IP-enabled applications work fine behind a Linux IP
  # Masquerade server.  But, some applications need a special
  # module to get their traffic in and out properly.
  #
  # Note: Some applications do NOT work well though a IP Masquerade server
  #       without special helper modules such as H.323-based programs.
  #       Please the IP-MASQ HOWTO for more details.
  #
  # Note #2: Only uncomment the modules that you REQUIRE to be loaded.
  #       The FTP module is loaded by default.
  #-------------------------------------------------------------------
  echo "  - Loading masquerading modules."

  #/sbin/modprobe ip_masq_cuseeme
  #/sbin/modprobe ip_masq_ftp
  #/sbin/modprobe ip_masq_irc
  #/sbin/modprobe ip_masq_quake
  #/sbin/modprobe ip_masq_raudio
  #/sbin/modprobe ip_masq_vdolive
  # If you downloaded and compiled the ICQ module from Section 5, use it
  #/sbin/modprobe ip_masq_icq
  # If you downloaded and compiled the H.323 module from Section 5, use it
  #/sbin/modprobe ip_masq_h323
  # If you downloaded and compiled the PPTP module from Section 5, use it
  #/sbin/insmod ip_masq_pptp
fi


#-------------------------------------------------------------------
# Default Policies
#-------------------------------------------------------------------
# Set all default policies to REJECT and flush all old rules.
#-------------------------------------------------------------------

# Change default policies to REJECT.
#
```

```
# We want to only EXPLICTITLY allow what traffic is allowed IN and OUT of the
# firewall.  All other traffic will be implicitly blocked.
#
echo "  - Set default policies to REJECT"
/sbin/ipchains -P input REJECT
/sbin/ipchains -P output REJECT
/sbin/ipchains -P forward REJECT

echo "  - Flushing all old rules and setting all default policies to REJECT "
# Flush all old rule sets
#
/sbin/ipchains -F input
/sbin/ipchains -F output
/sbin/ipchains -F forward


#********************************************************************
# Input Rules
#********************************************************************
echo "--------------------------------------------------------------------"
echo "Input Rules:"


# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then

  #-------------------------------------------------------------------
  # Incoming Traffic on the Internal LAN
  #-------------------------------------------------------------------
  # This section controls the INPUT traffic allowed to flow within the internal
  # LAN.  This means that all input traffic on the local network is valid.  If
  # you want to change this default setting and only allow certain types of
  # traffic within your internal network, you will need to comment this following
  # line and configure individual ACCEPT lines for each TCP/IP address you want
  # to let through.  A few example ACCEPT lines are provided below for
  # demonstration purposes.
  #
  # Sometimes it is useful to allow TCP connections in one direction but not the
  # other.  For example, you might want to allow connections to an external HTTP
  # server but not connections from that server.  The naive approach would be to
  # block TCP packets coming from the server. However, the better approach is to
  # use the -y flag which will block only the packets used to request a
  # connection.
  #-------------------------------------------------------------------
  echo "  - Setting input filters for traffic on the internal LAN."

  # DHCP Server.
  #
  # If you have configured a DHCP server on the Linux machine to serve IP
```

```
# addresses to the internal network, you will need to enable this section.
#
# This is an example of how to let input traffic flow through the local
# LAN if we have rejected all prior requests above.
#
# NOTE: Some distros change ipchains to NOT allow TCP connections for
#       DHCP.  Though TCP-based DHCP is really rare, it is part of
#       of the standard.
#
# Disabled by default
#echo "       Optional parameter: DHCPd server"
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p udp -s $UNIVERSE bootpc -d $BROADCAST/0 bootps
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $UNIVERSE bootpc -d $BROADCAST/0 bootps

# DMZ DHCPd - If we don't have a DMZ interface, dont do things for it
# #
# if [ "$INT2IF" != "" ]; then
#   #DMZ network
#   echo "       Optional parameter: Second INT2IF DHCPd server"
#   /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p udp -s $UNIVERSE bootpc -d $BROADCAST/0 bootps
#   /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $UNIVERSE bootpc -d $BROADCAST/0 bootps
# fi

#--------------------------------------------------------------------
# Explicit Access from Internal LAN Hosts
#--------------------------------------------------------------------
# This section is provided as an example of how to allow only SPECIFIC
# hosts on the internal LAN to access services on the firewall server.
# Many people might feel that this is extreme but many system attacks
# occur from the INTERNAL networks.
#
# Examples given allow access via FTP, FTP-DATA, SSH, and TELNET.
#
# In order for this rule set to work, you must first comment out the
# generic allow lines just above the final ALLOW HIGH PORTS at the END
# of this section.  That one line provides full access to the internal
# LAN by all internal hosts. You will then need to enable the lines
# below to allow any access at all.
#--------------------------------------------------------------------
#echo "  - Setting input filters for specific internal hosts."

# First allowed internal host to connect directly to the Linux server
#
# Disabled by default.
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ftp
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ftp-data
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ssh
#/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP telnet

# Second allowed internal host to connect directly to the Linux server
```

```
  #
  # Disabled by default.
  #/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ftp
  #/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ftp-data
  #/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ssh
  #/sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP telnet

  # This allows the ruleset to run if you use STATIC IPs and dont
  # enable DHCP
  echo "." > /dev/null

# End of the INTIF loop
fi

#---------------------------------------------------------------------
# Incoming Traffic from the External Interface
#---------------------------------------------------------------------
# This rule set will control specific traffic that is allowed in from
# the external interface.
#---------------------------------------------------------------------
#
echo "  - Setting input filters for traffic from the external interface."

# DHCP Clients.
#
# If you get a dynamic IP address for your ADSL or Cablemodem connection, you
# will need to enable these lines.
#
# NOTE: Some distros change ipchains to NOT allow TCP connections for
#       DHCP.  Though TCP-based DHCP is really rare, it is part of
#       of the standard.
#
# Enabled by default.
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p udp -s $UNIVERSE bootps -d $BROADCAST/0  bootpc
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE bootps -d $BROADCAST/0  bootpc

# FTP: Allow external users to connect to the Linux server ITSELF for
#      PORT-style FTP services.  This will NOT work for PASV FTP transfers.
#
# Disabled by default.
# echo "       Optional parameter: FTP server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP ftp
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP ftp-data

# IRCd:  Allow external users to connect to the Linux server ITSELF for
#        IRC services.
#
#        Make sure ircd is defined in /etc/services
#
# Disabled by default.
```

```
# echo "       Optional parameter: IRC server"
# /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP ircd

# HTTP: Allow external users to connect to the Linux server ITSELF for HTTP services.
#
# Disabled by default.
# echo "       Optional parameter: HTTP server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP http

# HTTPS: Allow external users to connect to the Linux server ITSELF for HTTPS services.
#
# Disabled by default.
# echo "       Optional parameter: HTTPS server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP https


# Advanced ICMP:  Some users prefer that their UNIX box NOT ping, etc.
#                 This is easy enough to do but be sure you know what you
#                 are doing.
#
#      There is an EXCELLENT paper on ICMP filtereing available at:
#
#    http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.0.pdf
#
#
#   NOTE:  When setting a FIREWALL to REJECT ICMP traffic, the resulting
#          reply traffic is automatically discarded per the RFCs
#
#   NOTE2: For a full list of all supported major and minor ICMP codes, run:
#              /sbin/ipchains -h icmp
#
# MOST are Disabled by default.
#
#
# Do NOT reply to ECHO REPLYs (type 0) from the Internet (this is NOT a
# good idea)
#
# echo "       Optional parameter: ICMP ECHO-REPLY inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type echo-reply $LOGGING
#
# Do NOT reply to TCP/UDP TRACEROUTE requests from the Internet (some find
# this useful)
#
# echo "       Optional parameter: TCP/UDP TRACEROUTE inbound filtered"
#
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 33434 $LOGGING
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP 33434 $LOGGING
#
# Do NOT reply to TRACEROUTE requests from the Internet (MS clients use
# ICMP ECHO and not TCP/UDP - some find this useful )
```

```
#
# echo "       Optional parameter: ICMP TRACEROUTE [for MS] inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type destination-unreachable $
#
# Do NOT reply to DESTINATION-UNREACHABLE (type 3) from the Internet (this
# is NOT a good idea - if you must do this then filter out the specific
# SUB-options such as PROTOCOL-UNREACHABLE in the OUTBOUND direction)
#
# echo "       Optional parameter: ICMP DESTINATION-UNREACHABLE inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type destination-unreachable $
#
# Do NOT reply to SOURCEQUENCH (type 4) from the Internet (this is NOT a
# good idea)
#
# echo "       Optional parameter: ICMP SOURCEQUENCH inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type source-quench $LOGGING
#
# Do NOT reply to ANY form of REDIRECT packets (type 5) (this can help
# stop OS fingerprinting)
#
echo "       Optional parameter: ICMP REDIRECT inbound filtered"
/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type redirect $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  echo "       Optional parameter: INT2IF - ICMP REDIRECT inbound filtered"
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP --icmp-type redirect $LOGGING
fi


# Do NOT allow PING requests (type 8) from the Internet (some find this
# useful)
#
# echo "       Optional parameter: ICMP ECHO inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type echo-request $LOGGING
#
# Do NOT reply to TTL-EXPIRED packets (type 11) from the Internet (this is
# NOT a good idea - do it OUTBOUND)
#
# echo "       Optional parameter: ICMP TTL-EXPIRED inbound filtered"
#/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type time-exceeded $LOGGING
#
# Do NOT reply to PARAMETER-PROBLEM packets (type 12) (this is NOT a good
# idea - filter this on OUTBOUND)
#
# echo "       Optional parameter: ICMP PARAMETER-PROBLEM inbound filtered"
# /sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type parameter-problem $LOGGI
#
```

```
# Do NOT reply to ICMP TIMESTAMP packets (type 13 and 14) (this can help
# stop OS fingerprinting)
#
echo "       Optional parameter: ICMP TIMESTAMP inbound filtered"
/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type timestamp-request $LOGGING
/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type timestamp-reply $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  echo "       Optional parameter: INT2IF - ICMP TIMESTAMP inbound filtered"
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP --icmp-type timestamp-request $LOG
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP --icmp-type timestamp-reply $LOGGI
fi


# ICMP INFORMATION (type 15 and 16) packet filtering is NOT supported by
# either LINUX or IPCHAINS (no big deal)
#
# Do NOT reply to ICMP ADDRESS MASK packets (type 17 and 18) (this can
# help stop OS fingerprinting)
#
echo "       Optional parameter: ICMP ADDRESS-MASK inbound filtered"
/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type address-mask-request $LOGG
/sbin/ipchains -A input -j REJECT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP --icmp-type address-mask-reply $LOGGIN


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  echo "       Optional parameter: INT2IF - ICMP ADDRESS-MASK inbound filtered"
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP --icmp-type address-mask-request $
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP --icmp-type address-mask-reply $LO
fi


# General ICMP: Allow ICMP packets from all external TCP/IP addresses.
#
# NOTE: Disabling ICMP packets via the firewall rule set can do far more
#       than just stop people from pinging your machine.  Many aspects of
#       TCP/IP and its associated applications rely on various ICMP
#       messages.  Without ICMP, both your Linux server and internal
#       Masq'ed computers might not work.
#
#   If you feel compelled to do ICMP filtering, do it by uncommenting your
#   desired traffic types from the section ABOVE and NOT here.
#
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p icmp -s $UNIVERSE -d $EXTIP
```

```
# DMZ ICMP - If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p icmp -s $UNIVERSE -d $INT2IP
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p icmp -s $INT2LAN -d $INTLAN
fi

# NNTP: Allow external computers to connect to the Linux server ITSELF
#       for NNTP (news) services.
#
# Disabled by default.
# echo "       Optional parameter: NNTP server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP nntp

# NTP: Allow external computers to connect to the Linux server ITSELF for
#      NTP (time) updates
#
#  NOTE:  Some NTP clients require TCP traffic.  Others require UDP.
#         Your pick!
#
# Disabled by default.
# echo "       Optional parameter: NTP server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP ntp
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP ntp

# TELNET: Allow external computers to connect to the Linux server ITSELF for
#         TELNET access.
#
# Disabled by default.
# echo "       Optional parameter: TELNET server"
#/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP telnet

# SSH server: Allow external computers to connect to the Linux server ITSELF
#             for SSH access.
#
# Disabled by default.
echo "       Optional parameter: SSH server"
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP ssh


#-------------------------------------------------------------------
# Specific Input Rejections on the EXTERNAL interface
#-------------------------------------------------------------------
# These rule sets reject specific traffic that you do not want into
# the system.
#-------------------------------------------------------------------
echo "  - Reject specific inputs."


# If we don't have an internal interface, dont do things for it
```

```
#
if [ "$INTIF" != "" ]; then
  # Remote interface, claiming to be local machines, IP spoofing, get lost & log
  /sbin/ipchains -A input -j REJECT -i $EXTIF -s $INTLAN -d $UNIVERSE $LOGGING
fi

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s $INTLAN -d $UNIVERSE $LOGGING
fi


# RFC1918 and IANA Reserved Address space Bogon filtering
#
# Filter all external traffic coming from either RESERVED or non-routed
# address space.
#
#  See ftp://ftp.iana.org/assignments/ipv4-address-space for up to date
#  results.
#
# Please run "whois IANA*@arin.net" and with a careful eye
# "whois RESERVED*@arin.net" for more info.
#
# -------------------------------------------------------------------
# NOTE *1*: Please notice that ALL IANA Reserved Address filters
#           (except for the Class-D and Class-E networks) have
#           been disabled as is seems that the IANA is releasing IP
#           address space without updating their tables.  There is
#           the email list called "bogon-announce" which you can
#           subscribe to here:
#                             http://www.cymru.com/Bogons/
#
# Note2: The bogon list changes ALL the time.  Unless you subscribe
#        to the above bogon list AND update your firewall when things
#        change, you will be blackholing traffic.
#
# Note3: that the address schemes from whois are silently using CLASSFULL
#        masks
#
# Note4: Some ISPs use RFC1918 addresses for internal addressing of
#         customers and keeping status on equipment.  Some customers of
#         General Instruments SURFboard cable modems might have similar
#         issues.
#
# -------------------------------------------------------------------


# Reserved-1
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 0.0.0.0/8 -d $UNIVERSE $LOGGING
```

```
# Reserved-9
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 1.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-2
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 2.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-5
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 5.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-7
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 7.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-10 and RFC1918 (10.x.x.x)
/sbin/ipchains -A input -j REJECT -i $EXTIF -s 10.0.0.0/8 -d $UNIVERSE $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s 10.0.0.0/8 -d $UNIVERSE $LOGGING
fi

# Reserved-23
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 23.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-27
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 27.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-31
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 31.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-36
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 36.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-37
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 37.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-39
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 39.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-42
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 42.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-74 and 75
# 74.0.0.0 - 75.55.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 74.0.0.0/7 -d $UNIVERSE $LOGGING

# Reserved-76 though 79
# 76.0.0.0 - 79.55.255.255
```

```
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 76.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved 89
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 89.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 90
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 90.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 91
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 91.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 92 though 95
# 92.0.0.0 - 95.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 92.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved 96 though 111
# 96.0.0.0 - 111.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 96.0.0.0/4 -d $UNIVERSE $LOGGING

# Reserved 112 though 119
# 112.0.0.0 - 119.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 112.0.0.0/5 -d $UNIVERSE $LOGGING

# Reserved 120 though 123
# 120.0.0.0 - 123.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 120.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved-127 127.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 127.0.0.0/8 -d $UNIVERSE $LOGGING

# BLACKHOLE3
#
# Disabled due to the fact that ALL reverse DNS functions (regardless of the
# address) will stop working properly.  If you have a good explination of
# why this is, I would love to hear it.
#
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 128.9.64.26/32 -d $UNIVERSE $LOGGING

# Includes NET-TEST-B
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 128.66.0.0/16 -d $UNIVERSE $LOGGING

# IANA-BBLK-RESERVED and RFC1918 (172.16-31.0.0)
/sbin/ipchains -A input -j REJECT -i $EXTIF -s 172.16.0.0/12 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s 172.16.0.0/12 -d $UNIVERSE $LOGGING
fi
```

```
# Reserved-173
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 173.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-174 through 175
# 174.0.0.0 - 175.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 174.0.0.0/7 -d $UNIVERSE $LOGGING

# Reserved-176 through 183
# 176.0.0.0 - 183.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 176.0.0.0/5 -d $UNIVERSE $LOGGING

# Reserved-184 through 187
# 184.0.0.0 - 187.255.255.255
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 184.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved-189
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 189.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-190
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 190.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-4
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 191.255.0.0/16 -d $UNIVERSE $LOGGING

# ROOT-NS-LAB - 192.0.0.0/24
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 192.0.0.0/24 -d $UNIVERSE $LOGGING

# NET-ROOTS-NS-LIVE - 192.0.1.0/24
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 192.0.1.0/24 -d $UNIVERSE $LOGGING

# NET-TEST - 192.0.2.0/24
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 192.0.2.0/24 -d $UNIVERSE $LOGGING

# RFC1918
#foo
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 192.168.0.0/16 -d $UNIVERSE $LOGGING

# RESERVED-13
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 197.0.0.0/16 -d $UNIVERSE $LOGGING

# Reserved-197
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 197.0.0.0/8 -d $UNIVERSE $LOGGING

# RESERVED-14
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 201.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-5
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 223.255.255.0/24 -d $UNIVERSE $LOGGING

# Reserved-223
```

```
#/sbin/ipchains -A input -j REJECT -i $EXTIF -s 223.0.0.0/24 -d $UNIVERSE $LOGGING

#Future use for Class-E:
/sbin/ipchains -A input -j REJECT -i $EXTIF -s 240.0.0.0/5 -d $UNIVERSE $LOGGING

#Future use for Class-F:
/sbin/ipchains -A input -j REJECT -i $EXTIF -s 248.0.0.0/5 -d $UNIVERSE $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s 240.0.0.0/5 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s 248.0.0.0/5 -d $UNIVERSE $LOGGING
fi


# -----------------
# Special Filtering
# -----------------


# Multicast:  Silently drop all multicast traffic for those users who
#             find this traffic filling up their logs.
#
# Disabled by default.
# echo "       Optional parameter: Ignore MULTICAST"
# /sbin/ipchains -A input -j REJECT -i $EXTIF -s $UNIVERSE -d 224.0.0.0/4


# NFS: Reject NFS traffic FROM and TO external machines.
#
# NOTE: NFS is one of the biggest security issues an administrator will face.
# Do NOT enable NFS over the Internet or any non-trusted networks unless you
# know exactly what you are doing.
#
# NOTE #2: the $LOGGING variable is NOT included here because if it was
#          enabled, your logs would grow too quickly to manage.
#
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 2049
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 2049 -d $EXTIP


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 2049
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 2049 -d $INT2IP
fi
```

```
# SMB and CIFS: Reject SMB and CIFS traffic FROM and TO external machines.
#
# NOTE: SMB (Win 3.x, 9x, NT) and CIFS (Win2k) is one of the biggest
#       security issues an administrator will face.  Do NOT enable SMB/CIFS
#       traffic to flow over the Internet or any non-trusted networks
#       unless you know exactly what you are doing.  If you NEED this
#       functionality, please use a IPSEC or PPTP VPN
#
# NOTE #2: the $LOGGING variable is NOT included here because if it was
#          enabled, your logs would grow too quickly to manage.
#
# Ports:   137 TCP/UDP (NetBIOS name service)
#          138 UDP     (NetBIOS datagram service) - TCP filtered just in case
#          139 TCP     (NetBIOS session service)  - UDP filtered just in case
#          445 TCP/UDP (MS CIFS in Win2k)

echo "     - Silently rejecting SMB and CIFS traffic on the external interface."
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 137
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP 137
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTBROAD 137
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTBROAD 137
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 138
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP 138
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTBROAD 138
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTBROAD 138
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 139
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP 139
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTBROAD 139
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTBROAD 139
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 445
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTIP 445
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE -d $EXTBROAD 445
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTBROAD 445
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 137 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE 137 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 138 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE 138 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 139 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE 139 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 445 -d $EXTIP
/sbin/ipchains -A input -j REJECT -i $EXTIF -p udp -s $UNIVERSE 445 -d $EXTIP

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 137
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2IP 137
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2BROAD 137
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2BROAD 137
```

```
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 138
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2IP 138
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2BROAD 138
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2BROAD 138
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 139
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2IP 139
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2BROAD 139
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2BROAD 139
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 445
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2IP 445
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE -d $INT2BROAD 445
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2BROAD 445
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 137 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE 137 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 138 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE 138 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 139 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE 139 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 445 -d $INT2IP
  /sbin/ipchains -A input -j REJECT -i $INT2IF -p udp -s $UNIVERSE 445 -d $INT2IP
fi


#------------------------------------------------------------------
# Incoming Traffic on all Interfaces
#------------------------------------------------------------------
# This will control input traffic for all interfaces.  This is
# usually used for what could be considered as public services.
#------------------------------------------------------------------
echo "  - Setting input filters for public services [all interfaces]."

# AUTH: Allow the authentication protocol, ident, to function on all
#       interfaces but disable it in /etc/inetd.conf.  The reason to
#       allow this traffic in but block it via Inetd is because some
#       legacy TCP/IP stacks don't deal with REJECTed "auth" requests
#       properly.
#
# Traffic TO your machine and FROM your machine
/sbin/ipchains -A input -j ACCEPT -p tcp -s $UNIVERSE -d $UNIVERSE auth
/sbin/ipchains -A input -j ACCEPT -p tcp -s $UNIVERSE auth -d $UNIVERSE

# BOOTP/DHCP: Reject all stray bootp traffic.
#
# Disabled by default.
#/sbin/ipchains -A input -j REJECT -p udp -s $UNIVERSE bootpc

# DNS: If you are running an authoritative DNS server, you must open
#      up the DNS ports on all interfaces to allow lookups.  If you are
#      running a caching DNS server, you will need to at least open the DNS
#      ports to internal interfaces.
```

```
#
#      It is recommend to secure DNS by restricting zone transfers and split
#      DNS servers as documented in Step 4.
#
# Disabled by default.
#echo "       Optional parameter: DNS server"
#/sbin/ipchains -A input -j ACCEPT -p tcp -s $UNIVERSE -d $UNIVERSE domain
#/sbin/ipchains -A input -j ACCEPT -p udp -s $UNIVERSE -d $UNIVERSE domain

# RIP: Reject all stray RIP traffic.  Many improperly configured
#      networks propagate network routing protocols to the edge of the
#      network.  The follow line will allow you explicitly filter it here
#      without logging to SYSLOG.
#
# Disabled by default.
#/sbin/ipchains -A input -j REJECT -p udp -s $UNIVERSE -d $UNIVERSE route

# SMTP: If this server is an authoritative SMTP email server, you must
#       allow SMTP traffic to all interfaces.
#
# Disabled by default.
#echo "       Optional parameter: SMTP server"
#/sbin/ipchains -A input -j ACCEPT -p tcp -s $UNIVERSE -d $UNIVERSE smtp

# SQUID Proxy w/ JunkBuster
#
# If you are using Squid w/ Junkbuster enabled [Banner filtering],  you will
# need to enable the following lines to do the IPCHAINS port redirection to
# port 3128.  This also assumes that you have Squid properly configured and
# running.
#
# Disabled by default.
#echo "       Optional parameter: SQUID transparent proxy"
#/sbin/ipchains -A input -j ACCEPT -i $LOOPBACKIF -p tcp -d $LOOPBACKIP/32 www
#
# If we don't have an internal interface, dont do things for it
#
#if [ "$INTIF" != "" ]; then
#  /sbin/ipchains -A input -j ACCEPT -i $INTIF -p tcp -s $INTLAN -d $INTIP/32 www
#  /sbin/ipchains -A input -j REDIRECT 3128 -i $INTIF -p tcp -s $INTLAN -d $INTLAN www $LOGGING
#fi

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  # DMZ network - Enable this section if you have a wireless segment
  #
  # Enabled by default if INT2IF is valid
  echo "       Optional parameter: DMZ segment - SSH"
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $INT2LAN ssh -d $UNIVERSE
```

```
  # Enabled by default if INT2IF is valid
  echo "       Optional parameter: DMZ segment - DNS"
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $INT2LAN -d $UNIVERSE domain
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p udp -s $INT2LAN -d $UNIVERSE domain

  #Enable this option if you want ALL DMZ machines to access all network services
  # on all interfaces.  The alternative is allow host by host access in the
  # DMZ SecureHOST section below
  #
  # Disabled by default.
  #/sbin/ipchains -A input -j ACCEPT -i $INT2IF -s $INT2LAN -d $UNIVERSE
fi



#--------------------------------------------------------------------
# Specific Input Rejections from ANY interface
#--------------------------------------------------------------------
# These rule sets reject specific traffic that you do not want out of
# the system.
#--------------------------------------------------------------------
#echo "  - Reject traffic for specific domains."

# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  #Do not allow ANY internal hosts to be able to reach the following sites:
  #
  #Disabled by default.

  #The Doubleclick example will filter ALL types of traffic to the given
  #       class-C networks including WWW, SMTP(email, etc traffic.  If you
  #     want a slightly less restrictive example, see the AOL example.
  #
  #Doubleclick.net and .com is renowned for their WWW ad banners
  #
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 63.160.54.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 128.11.92.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 199.95.206.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 199.95.207.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 199.95.208.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 199.95.210.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 204.178.112.160/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 204.253.104.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 208.10.202.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 208.203.243.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 208.211.225.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 208.228.86.0/24
  #/sbin/ipchains -A input -j REJECT -i $INTIF -p tcp -s $UNIVERSE -d 209.67.38.0/24
```

```
  #This is required to complete the if..then loop
  echo "." > /dev/null
fi

#AOL.com is renowned for their users sending SPAM to millions of people on
#        the Inet.  Though you might want to filter email from them, you
#          might want to still be able to go look at some of their their
#          WWW pages.  This example ONLY filters EMAIL and nothing else.
#
#/sbin/ipchains -A input -j REJECT -p tcp -s $UNIVERSE 25 -d 152.163.159.0/24
#/sbin/ipchains -A input -j REJECT -p tcp -s $UNIVERSE 25 -d 205.188.157.0/24



#-------------------------------------------------------------------
# Explicit INPUT Access from external LAN Hosts
#-------------------------------------------------------------------
# This controls external access from specific external hosts (secure hosts).
# This example permits FTP, FTP-DATA, SSH, POP-3 and TELNET traffic from a
# secure host INTO the firewall. In addition to these input rules, we must also
# explicitly allow the traffic from the remote host to get out.  See the rules
# in the output section for more details
#
# Disabled as default.
#-------------------------------------------------------------------
echo "  - SECUREHOST: Setting input filters for explicit hosts."

# The secure host section

if [ "$SECUREHOST" != "" ]; then
  echo "     * Allowing $SECUREHOST INPUT for ftp, ftp-data, ssh"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST -d $EXTIP ssh
fi

if [ "$SECUREHOST2" != "" ]; then
  echo "     * Allowing $SECUREHOST2 INPUT for ftp, ftp-data, ssh, www, telnet, imap"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP ssh
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP telnet
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP www
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST2 -d $EXTIP imap
fi

if [ "$SECUREHOST3" != "" ]; then
  echo "     * Allowing $SECUREHOST3 INPUT for ftp, ftp-data, ssh, www"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST3 -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST3 -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST3 -d $EXTIP ssh
```

```
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST3 -d $EXTIP www
fi

if [ "$SECUREHOST4" != "" ]; then
  echo "     * Allowing $SECUREHOST4 INPUT for ftp, ftp-data, ssh"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ssh
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP www
fi

if [ "$SECUREHOST5" != "" ]; then
  echo "     * Allowing $SECUREHOST5 INPUT for ftp, ftp-data, ssh, www"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST5 -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST5 -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST5 -d $EXTIP ssh
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST5 -d $EXTIP www
fi

if [ "$SECUREHOST6" != "" ]; then
  echo "     * Allowing $SECUREHOST6 INPUT for ftp, ftp-data, ssh, pop-3, and telnet"
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ftp
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ftp-data
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP ssh
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP pop-3
  /sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST4 -d $EXTIP telnet
fi


echo "  - DMZ-SECUREHOST: Setting input filters for explicit hosts."
# If we don't have a DMZ interface, dont do things for it
#
if ( [ "$INT2IF" != "" ] && [ "$DMZHOST1" != "" ] ); then
   #DMZ SecureHost
   #
   echo "     * Allowing $DMZHOST1 INPUT for ssh to the Linux server and the INET"
   /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST1 -d $INT2IP ssh
   /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST1 -d $INTLAN ssh
   /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST1 -d $UNIVERSE
fi

if ( [ "$INT2IF" != "" ] && [ "$DMZHOST2" != "" ] ); then
  echo "     * Allowing $DMZHOST2 INPUT for ssh to the Linux server and the INET"
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST2 -d $INT2IP ssh
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST2 -d $INTLAN ssh
  /sbin/ipchains -A input -j ACCEPT -i $INT2IF -p tcp -s $DMZHOST2 -d $UNIVERSE
fi


if [ "$INT2IF" != "" ]; then
```

```
  #DMZ network - this is where most of the wireless filtering occurs
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s $INT2LAN -d $INTLAN $LOGGING
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s $INT2LAN -d $INT2LAN $LOGGING
  /sbin/ipchains -A input -j REJECT -i $INT2IF -s $INTLAN -d $UNIVERSE $LOGGING
fi


# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  # Allow ALL internal interfaces to access the Inet
  # ------------------------------------------------
  # Local interface, local machines, going anywhere is valid.
  #
  # The main reason why this is at the BOTTOM of the INPUT section is to
  # make sure that all required DENY/REJECT firewall lines are hit before
  # allowing all internal traffic.  If you DON'T want to allow ALL internal
  # traffic to get out to the Internet, put a "#" in the
  # front of the line below and un-#ed out the lines at the top of this
  # section to allow only specific internal HOSTS to get out.
  #
  # Comment this line out if you want to only allow specific traffic on the
  # internal network.
  /sbin/ipchains -A input -j ACCEPT -i $INTIF -s $INTLAN -d $UNIVERSE
fi

# Loopback interface is valid.
#
/sbin/ipchains -A input -j ACCEPT -i $LOOPBACKIF -s $UNIVERSE -d $UNIVERSE


# HIGH PORTS:
#
# Enable all high unprivileged ports for all reply TCP/UDP traffic
#
# NOTE: The use of the "! -y" flag filters TCP traffic that doesn't have the
#       SYN bit set.  In other words, this means that any traffic that is
#       trying to initiate traffic to your server on a HIGH port will be
#       rejected.
#
#       The only HIGH port traffic that will be accepted is either return
#       traffic that the server originally initiated or UDP-based traffic.
#
# NOTE2: Please note that port 20 for ACTIVE FTP sessions should NOT use
#        SYN filtering.  Because of this, we must specifically allow it in.
#
echo "  - Enabling all input REPLY [TCP/UDP] traffic on high ports."
/sbin/ipchains -A input -j ACCEPT ! -y -p tcp -s $UNIVERSE -d $EXTIP $UNPRIVPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $UNIVERSE ftp-data -d $EXTIP $UNPRIVPORTS
/sbin/ipchains -A input -j ACCEPT -p udp -s $UNIVERSE -d $EXTIP $UNPRIVPORTS
```

```
# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  #DMZ network and removed FTP as it is insecure
  /sbin/ipchains -A input -j ACCEPT ! -y -p tcp -s $UNIVERSE -d $INT2IP $UNPRIVPORTS
  /sbin/ipchains -A input -j ACCEPT -p udp -s $UNIVERSE -d $INT2IP $UNPRIVPORTS
fi

#----------------------------------------------------------------------
# Catch All INPUT Rule
#----------------------------------------------------------------------
#
echo "  - Final input catch all rule."

# All other incoming is denied and logged.
/sbin/ipchains -A input -j REJECT -s $UNIVERSE -d $UNIVERSE $LOGGING


#**********************************************************************
# Output Rules
#**********************************************************************
echo "----------------------------------------------------------------------"
echo "Output Rules:"

#----------------------------------------------------------------------
# Outgoing Traffic on the Internal LAN
#----------------------------------------------------------------------
# This rule set provides policies for traffic that is going out on the internal
# LAN.
#
# In this example, all traffic is allowed out.  Therefore there is no
# requirement to implement individual filters.  However, as with the input
# section above, examples are given for demonstrative purposes.  It is also
# noted that the same rules, outlined above, apply regarding the order of the
# filtering rules.
#----------------------------------------------------------------------
echo "  - Setting output filters for traffic on the internal LAN."

# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  # Local interface, any source going to local net is valid.
  /sbin/ipchains -A output -j ACCEPT -i $INTIF -s $UNIVERSE -d $INTLAN
fi

# Loopback interface is valid.
/sbin/ipchains -A output -j ACCEPT -i $LOOPBACKIF -s $UNIVERSE -d $UNIVERSE

# If we don't have an internal interface, dont do things for it
```

```
#
if [ "$INTIF" != "" ]; then
  # DHCP: If you have configured a DHCP server on this Linux machine, you
  #       will need to enable the following rule set.
  #
  # NOTE: Some distros change ipchains to NOT allow TCP connections for
  #       DHCP.  Though TCP-based DHCP is really rare, it is part of
  #       of the standard.
  #
  # Enabled by default.
  echo "       Optional parameter: DHCPd server"
  /sbin/ipchains -A output -j ACCEPT -i $INTIF -p udp -s $INTIP/32 bootps -d $BROADCAST/0 bootpc
  /sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $INTIP/32 bootps -d $BROADCAST/0 bootpc

  #If you DISABLE the lines above, you need this following line to
  #let the if..then statement run without failing out
  echo "." > /dev/null
fi

# DMZ DHCP server - If we don't have a DMZ interface, dont do things for it
#
# Disabled by default
#
# if [ "$INT2IF" != "" ]; then
#  /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p udp -s $INT2IP/32 bootps -d $BROADCAST/0 bootpc
#  /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INT2IP/32 bootps -d $BROADCAST/0 bootpc
# fi


# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  # HTTP: The following is an example of how to allow HTTP traffic to an
  #       intranet WWW server without allowing access from the external
  #       network.
  #
  # Disabled by default.
  # echo "       Optional parameter: WWW server"
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $INTIP/32 http -d $INTLAN


  # APC PowerChute for Linux:  The following is needed for APCs PowerChute
  #       software for Linux.  The way it works is that it broadcasts the
  #       private network looking for the upsd daemon.
  #
  # Disabled by default.
  #echo "       Optional parameter: UPSd server"
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p udp -s $INTIP/32 -d $BROADCAST 5456

  #This is required to complete the if..then loop if it is empty
```

```
  echo "." > /dev/null
fi


# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  #----------------------------------------------------------------------
  # Explicit Output from Internal LAN Hosts
  #----------------------------------------------------------------------
  # The following rule sets only allow SPECIFIC hosts on the internal LAN to
  # access services on this firewall server itself.  Many people might feel that
  # this is extreme but many system attacks occur from the INTERNAL network as
  # well.
  #
  # Examples given allow access via FTP, FTP-DATA, SSH, and TELNET.
  #
  # In order for this rule set to work, you must first comment out the line above
  # that provides full access to the internal LAN by all internal hosts.
  #
  # Disabled by default.
  #----------------------------------------------------------------------
  #echo "  - Setting output filters for specific internal hosts."

  # First host
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ftp
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ftp-data
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP ssh
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST1IP -d $INTIP telnet

  # Second host
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ftp
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ftp-data
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP ssh
  #/sbin/ipchains -A output -j ACCEPT -i $INTIF -p tcp -s $HOST2IP -d $INTIP telnet

  #This is required to complete the if..then loop if it is empty
  echo "." > /dev/null
fi

#----------------------------------------------------------------------
# Outgoing Traffic on the External Interface
#----------------------------------------------------------------------
# This rule set will control what traffic can go out on the external interface.
#----------------------------------------------------------------------
echo "  - Setting input filters for traffic to the external interface."

# DHCP Client: If your Linux server is connected via DSL or a Cablemodem
#              connection and you get dynamic DHCP addresses, you will need to
#              enable the following rule sets.
```

```
#
# NOTE: Some distros change ipchains to NOT allow TCP connections for
#       DHCP.  Though TCP-based DHCP is really rare, it is part of
#       of the standard.
#
# Enabled by default.
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $UNIVERSE bootpc -d $UNIVERSE bootps
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p udp -s $UNIVERSE bootpc -d $UNIVERSE bootps

# FTP: Allow FTP traffic (the Linux server is a FTP server)
#
# Disabled by default.
# echo "       Optional parameter: FTP server"
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $UNIVERSE
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $UNIVERSE

# IRCd: Allow IRC traffic (the Linux server is a IRC server)
#
#        Make sure ircd is defined in /etc/services
#
# Disabled by default
# echo "       Optional parameter: IRC server"
# /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ircd -d $UNIVERSE

# HTTP: Allow HTTP traffic (the Linux server is a WWW server)
#
# Disabled by default
# echo "       Optional parameter: WWW server"
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP http -d $UNIVERSE

# HTTPS: Allow HTTPS traffic (the Linux server is a WWW server)
#
# Disabled by default
# echo "       Optional parameter: HTTPS server"
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP https -d $UNIVERSE

# NTP: Allow NTP updates (the Linux server is a NTP server)
#
#  NOTE:  Some NTP clients require TCP traffic.  Others require UDP.
#         Your pick!
#
# Disabled by default
# echo "       Optional parameter: NTP server"
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ntp -d $UNIVERSE
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p udp -s $EXTIP ntp -d $UNIVERSE

# TELNET: Allow telnet traffic (the Linux server is a TELNET server)
#
# Disabled by default
# echo "       Optional parameter: TELNET server"
```

```
#/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP telnet -d $UNIVERSE

# SSH server: Allow outgoing SSH traffic (the Linux server is a SSH server)
#
# Disabled by default
#
# echo "       Optional parameter: SSH server"
# /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $UNIVERSE


#-------------------------------------------------------------------
# Outgoing Traffic on all Interfaces
#-------------------------------------------------------------------
# This will control output traffic for all interfaces.  This is
# usually used for what could be considered as public services.  It
# is noted that we provide a few rejection rule sets as examples but
# these are not required due to the overall REJECT statement above.
#-------------------------------------------------------------------
echo "  - Setting output filters for public services on all interfaces."

# AUTH: Allow the authentication protocol, ident, to function on all
#       interfaces but disable it in /etc/inetd.conf.  The reason to
#       allow this traffic in but block it via Inetd is because some
#       legacy TCP/IP stacks don't deal with REJECTed "auth" requests
#       properly.
#
# Traffic TO your machine and FROM your machine
/sbin/ipchains -A output -j ACCEPT -p tcp -s $UNIVERSE auth -d $UNIVERSE
/sbin/ipchains -A output -j ACCEPT -p tcp -s $UNIVERSE -d $UNIVERSE auth

# DNS: If you your Linux server is an authoritative DNS server, you must
# enable this rule set
#
# Disabled by default
#echo "       Optional parameter: DNS server"
#/sbin/ipchains -A output -j ACCEPT -p tcp -s $EXTIP domain -d $UNIVERSE
#/sbin/ipchains -A output -j ACCEPT -p udp -s $EXTIP domain -d $UNIVERSE


# Advanced ICMP:  Some users prefer that their UNIX box NOT ping, etc.
#                 This is easy enough to do but be sure you know what you
#                 are doing.
#
#      There is an EXCELLENT paper on ICMP filtereing available at:
#
#    http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.0.pdf
#
#
#   NOTE:  When setting a FIREWALL to REJECT ICMP traffic, the resulting
#          reply traffic is automatically discarded per the RFCs
```

```
#
#   NOTE2: For a full list of all supported major and minor ICMP codes, run:
#              /sbin/ipchains -h icmp
#
# MOST are Disabled by default.
#
#
# Do NOT reply to ICMP ECHO REPLYs (type 0) requests from the Internet
#   (some find this useful)
#
# echo "       Optional parameter: ICMP ECHO REPLY outbound filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type echo-reply
#
# Do NOT reply to TCP/UDP TRACEROUTE requests from the Internet (some find
#   this useful)
#
# echo "       Optional parameter: TCP/UDP TRACEROUTE outbound filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 33434 $LOGGING
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 33434 $LOGGING
#
# Do NOT reply to TRACEROUTE requests from the Internet (MS clients use
#   ICMP ECHOs instead of TCP/UDP - some find this useful )
#
# echo "       Optional parameter: ICMP TRACEROUTE [MS] outbound filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type echo-request $LOGGING
#
# Do NOT reply to DESTINATION-UNREACHABLE (type 3) from the Internet (this
#   is NOT a good idea - if you must do this then filter out the specific
#   SUB-options such as PROTOCOL-UNREACHABLE in the OUTBOUND direction)
#
# echo "       Optional parameter: ICMP DESTINATION-UNREACHABLE output filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type destination-unreachable
#
# Do NOT reply to SOURCEQUENCH (type 4) from the Internet (this is NOT a
#   good idea)
#
# echo "       Optional parameter: ICMP SOURCEQUENCH outbound filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type source-quench $LOGGING
#
# Do NOT reply to ANY form of ICMP REDIRECT packets (type 5) (this can
#   help stop OS fingerprinting)
#
echo "       Optional parameter: ICMP REDIRECT outbound filtered"
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type redirect $LOGGING

# If we don’t have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type redirect $LOGGING
fi
```

```
# Do NOT allow PING requests (type 8) from the Internet (some find this
#   useful)
#
# echo "       Optional parameter: ICMP ECHO outbound filtered"
#/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type echo-request $LOGGING
#
# Do NOT reply to TTL-EXPIRED packets (type 11) from the Internet (this
#   is NOT a good idea - do it OUTBOUND)
#
echo "       Optional parameter: ICMP TTL-EXPIRED outbound filtered"
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type ttl-zero-during-reassembl


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type ttl-zero-during-reass
fi


# Do NOT reply to PARAMETER-PROBLEM packets (type 12) (this is NOT a good
#   idea - filter this on OUTBOUND)
#
echo "       Optional parameter: ICMP PARAMETER-PROBLEM outbound filtered"
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type parameter-problem $LOGGIN


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type parameter-problem $LO
fi


# Do NOT reply to ICMP TIMESTAMP packets (type 13 and 14) (this can help
#   stop OS fingerprinting)
#
echo "       Optional parameter: ICMP TIMESTAMP outbound filtered"
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type timestamp-request $LOGGIN
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type timestamp-reply $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type timestamp-request $LO
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type timestamp-reply $LOGG
fi
```

```
# ICMP INFORMATION (type 15 and 16) packet filtering is NOT supported by
#   either LINUX or IPCHAINS (no big deal)
#
# Do NOT reply to ICMP ADDRESS MASK packets (type 17 and 18) (this can help
#   stop OS fingerprinting)
#
echo "       Optional parameter: ICMP ADDRESS-MASK outbound filtered"
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type address-mask-request $LOG
/sbin/ipchains -A output -j REJECT -i $EXTIF -p icmp -s $EXTIP -d $UNIVERSE --icmp-type address-mask-reply $LOGGI

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type address-mask-request
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p icmp -s $INT2IP -d $UNIVERSE --icmp-type address-mask-reply $L
fi


# General ICMP: Allow ICMP traffic out
#
# NOTE: Disabling ICMP packets via the firewall rule set can do far
# more than just stop people from pinging your machine.  Many aspects
# of TCP/IP and its associated applications rely on various ICMP
# messages.  Without ICMP, both your Linux server and internal Masq'ed
# computers might not work.
#
#   If you feel compelled to do ICMP filtering, do it by uncommenting your
#   desired traffic types from the section ABOVE and NOT here.
#
/sbin/ipchains -A output -j ACCEPT -p icmp -s $UNIVERSE -d $UNIVERSE


# NNTP: This allows NNTP-based news out.
#
# Disabled by default
# echo "       Optional parameter: NNTP server"
#/sbin/ipchains -A output -j ACCEPT -p tcp -s $EXTIP nntp -d $UNIVERSE

# SMTP: If the Linux servers is either an authoritative SMTP server or
# relay, you must allow this rule set.
#
# Disabled by default
#echo "       Optional parameter: SMTP server"
#/sbin/ipchains -A output -j ACCEPT -p tcp -s $EXTIP smtp -d $UNIVERSE


#--------------------------------------------------------------------
# Output to Explicit Hosts
#--------------------------------------------------------------------
```

```
# This controls output to specific external hosts [secure hosts].  This example
# implementation allows ssh and pop-3 protocols out to the secure host.  In
# addition to these rules, we must also explicitly allow the traffic in from
# the remote host.  See the input rules above to see this take place.
#
# Disabled by default.
#---------------------------------------------------------------------
echo "  - SECUREHOST: Setting output filters for explicit hosts."

# The secure host
#
if [ "$SECUREHOST" != "" ]; then
   echo "     * Allowing $SECUREHOST OUTPUT for ftp, ftp-data, ssh"
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $SECUREHOST $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $SECUREHOST $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $SECUREHOST $UNPRIVPORTS
fi

if [ "$SECUREHOST2" != "" ]; then
   echo "     * Allowing $SECUREHOST2 OUTPUT for ftp, ftp-data, ssh, telnet, imap, and www"
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $SECUREHOST2 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $SECUREHOST2 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $SECUREHOST2 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP telnet -d $SECUREHOST2 $UNPRIVPORT
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP www -d $SECUREHOST2 $UNPRIVPORT
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP imap -d $SECUREHOST2 $UNPRIVPORT
fi

if [ "$SECUREHOST3" != "" ]; then
   echo "     * Allowing $SECUREHOST3 OUTPUT for ftp, ftp-data, ssh, www"
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $SECUREHOST3 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $SECUREHOST3 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $SECUREHOST3 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP www -d $SECUREHOST3 $UNPRIVPORTS
fi

if [ "$SECUREHOST4" != "" ]; then
   echo "     * Allowing $SECUREHOST4 OUTPUT for ftp, ftp-data, ssh, www"
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $SECUREHOST4 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $SECUREHOST4 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $SECUREHOST4 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP www -d $SECUREHOST4 $UNPRIVPORTS
fi

if [ "$SECUREHOST5" != "" ]; then
   echo "     * Allowing $SECUREHOST5 OUTPUT for ftp, ftp-data, ssh, www"
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp -d $SECUREHOST5 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ftp-data -d $SECUREHOST5 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP ssh -d $SECUREHOST5 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP www -d $SECUREHOST5 $UNPRIVPORTS
```

```
fi

echo "  - DMZ-SECUREHOST: Setting output filters for explicit hosts."
# If we don't have a DMZ interface, dont do things for it
#
if ( [ "$INT2IF" != "" ] && [ "$DMZHOST1" != "" ] ); then
   echo "     * Allowing $DMZHOST1 OUTPUT for ssh, ftp"
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INT2IP ftp -d $DMZHOST1 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INTLAN ssh -d $DMZHOST1 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INTLAN $UNPRIVPORTS -d $DMZHOST1 ssh
fi

if ( [ "$INT2IF" != "" ] && [ "$DMZHOST2" != "" ] ); then
   echo "     * Allowing $DMZHOST2 OUTPUT for ssh, ftp"
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INT2IP ftp -d $DMZHOST2 $UNPRIVPORTS
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INTLAN $UNPRIVPORTS -d $DMZHOST2 ssh
   /sbin/ipchains -A output -j ACCEPT -i $INT2IF -p tcp -s $INTLAN ssh -d $DMZHOST2 $UNPRIVPORTS
 fi

#---------------------------------------------------------------------
# Specific Output Rejections
#---------------------------------------------------------------------
# These rule sets reject specific traffic that you do not want out of
# the system.
#---------------------------------------------------------------------
echo "  - Reject specific outputs."

# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  # Reject outgoing traffic to the local net from the remote interface,
  # stuffed routing; deny & log
  /sbin/ipchains -A output -j REJECT -i $EXTIF -s $UNIVERSE -d $INTLAN $LOGGING
fi

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s $UNIVERSE -d $INTLAN $LOGGING
fi

# If we don't have an internal interface, dont do things for it
#
if [ "$INTIF" != "" ]; then
  # Reject outgoing traffic from the local net from the external interface,
  # stuffed masquerading, deny and log
  /sbin/ipchains -A output -j REJECT -i $EXTIF -s $INTLAN -d $UNIVERSE $LOGGING
fi

# If we don't have a DMZ interface, dont do things for it
```

```
#
if [ "$INT2IF" != "" ]; then
  #DMZ network - block all outgoing DMZ traffic unless allowed somewhere above
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s $INTLAN -d $UNIVERSE $LOGGING
fi



# RFC1918 and IANA Reserved Address space Bogon filtering
#
# Filter all external traffic coming from either RESERVED or non-routed
# address space.
#
#  See ftp://ftp.iana.org/assignments/ipv4-address-space for up to date
#  results.
#
# Please run "whois IANA*@arin.net" and with a careful eye
# "whois RESERVED*@arin.net" for more info.
#
# --------------------------------------------------------------------
# NOTE *1*: Please notice that ALL IANA Reserved Address filters
#           (except for the Class-D and Class-E networks) have
#           been disabled as is seems that the IANA is releasing IP
#           address space without updating their tables.  There is
#           the email list called "bogon-announce" which you can
#           subscribe to here:
#                            http://www.cymru.com/Bogons/
#
# Note2: The bogon list changes ALL the time.  Unless you subscribe
#        to the above bogon list AND update your firewall when things
#        change, you will be blackholing traffic.
#
# Note3: that the address schemes from whois are silently using CLASSFULL
#        masks
#
# Note4: Some ISPs use RFC1918 addresses for internal addressing of
#         customers and keeping status on equipment.  Some customers of
#         General Instruments SURFboard cable modems might have similar
#         issues.
#
# --------------------------------------------------------------------


# Reserved-1
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 0.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-9
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 1.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-2
```

```
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 2.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-5
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 5.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-7
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 7.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-10 and RFC1918 (10.x.x.x)
/sbin/ipchains -A output -j REJECT -i $EXTIF -s 10.0.0.0/8 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s 10.0.0.0/8 -d $UNIVERSE $LOGGING
fi

# Reserved-23
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 23.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-27
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 27.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-31
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 31.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-36
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 36.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-37
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 37.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-39
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 39.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-42
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 42.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-74 and 75
# 74.0.0.0 - 75.55.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 74.0.0.0/7 -d $UNIVERSE $LOGGING

# Reserved-76 though 79
# 76.0.0.0 - 79.55.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 76.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved 89
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 89.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 90
```

```
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 90.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 91
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 91.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved 92 though 95
# 92.0.0.0 - 95.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 92.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved 96 though 111
# 96.0.0.0 - 111.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 96.0.0.0/4 -d $UNIVERSE $LOGGING

# Reserved 112 though 119
# 112.0.0.0 - 119.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 112.0.0.0/5 -d $UNIVERSE $LOGGING

# Reserved 120 though 123
# 120.0.0.0 - 123.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 120.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved-127 127.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 127.0.0.0/8 -d $UNIVERSE $LOGGING

# BLACKHOLE3
#
# Disabled due to the fact that ALL reverse DNS functions (regardless of the
# address) will stop working properly.  If you have a good explination of
# why this is, I would love to hear it.
#
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 128.9.64.26/32 -d $UNIVERSE $LOGGING

# Includes NET-TEST-B
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 128.66.0.0/16 -d $UNIVERSE $LOGGING

# IANA-BBLK-RESERVED and RFC1918 (172.16-31.0.0)
/sbin/ipchains -A output -j REJECT -i $EXTIF -s 172.16.0.0/12 -d $UNIVERSE $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s 172.16.0.0/12 -d $UNIVERSE $LOGGING
fi

# Reserved-173
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 173.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-174 through 175
# 174.0.0.0 - 175.255.255.255
```

```
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 174.0.0.0/7 -d $UNIVERSE $LOGGING

# Reserved-176 through 183
# 176.0.0.0 - 183.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 176.0.0.0/5 -d $UNIVERSE $LOGGING

# Reserved-184 through 187
# 184.0.0.0 - 187.255.255.255
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 184.0.0.0/6 -d $UNIVERSE $LOGGING

# Reserved-189
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 189.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-190
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 190.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-4
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 191.255.0.0/16 -d $UNIVERSE $LOGGING

# ROOT-NS-LAB - 192.0.0.0/24
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 192.0.0.0/24 -d $UNIVERSE $LOGGING

# NET-ROOTS-NS-LIVE - 192.0.1.0/24
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 192.0.1.0/24 -d $UNIVERSE $LOGGING

# NET-TEST - 192.0.2.0/24
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 192.0.2.0/24 -d $UNIVERSE $LOGGING

# RFC1918
#foo
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 192.168.0.0/16 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j ACCEPT -i $INT2IF -s $UNIVERSE -d $INT2LAN
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s $UNIVERSE -d 192.168.0.0/16 $LOGGING
fi

# RESERVED-13
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 197.0.0.0/16 -d $UNIVERSE $LOGGING

# Reserved-197
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 197.0.0.0/8 -d $UNIVERSE $LOGGING

# RESERVED-14
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 201.0.0.0/8 -d $UNIVERSE $LOGGING

# Reserved-5
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 223.255.255.0/24 -d $UNIVERSE $LOGGING
```

```
# Reserved-223
#/sbin/ipchains -A output -j REJECT -i $EXTIF -s 223.0.0.0/24 -d $UNIVERSE $LOGGING

#Future use for Class-E:
/sbin/ipchains -A output -j REJECT -i $EXTIF -s 240.0.0.0/5 -d $UNIVERSE $LOGGING

#Future use for Class-F:
/sbin/ipchains -A output -j REJECT -i $EXTIF -s 248.0.0.0/5 -d $UNIVERSE $LOGGING


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s 240.0.0.0/5 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -s 248.0.0.0/5 -d $UNIVERSE $LOGGING
fi


# -----------------
# Special Filtering
# -----------------

# Multicast:  Silently drop all multicast traffic for those users who
#             find this traffic filling up their logs.
#
# Disabled by default.
# echo "       Optional parameter: Ignore MULTICAST"
# /sbin/ipchains -A output -j REJECT -i $EXTIF -s $UNIVERSE -d 224.0.0.0/4


# NFS: Reject NFS traffic FROM and TO external machines.
#
# NOTE: NFS is one of the biggest security issues an administrator will face.
# Do NOT enable NFS over the Internet or any non-trusted networks unless you
# know exactly what you are doing.
#
# NOTE #2: the $LOGGING variable is NOT included here because if it was
#          enabled, your logs would grow too quickly to manage.
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $UNIVERSE -d $EXTIP 2049
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $UNIVERSE 2049 -d $EXTIP


# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $UNIVERSE -d $INT2IP 2049
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $UNIVERSE 2049 -d $INT2IP
fi
```

```
# SMB and CIFS: Reject SMB and CIFS traffic FROM external machines.
#
# NOTE: SMB (Win 3.x, 9x, NT) and CIFS (Win2k) is one of the biggest
#       security issues an administrator will face.  Do NOT enable SMB/CIFS
#       traffic to flow over the Internet or any non-trusted networks
#       unless you know exactly what you are doing.  If you NEED this
#       functionality, please use a IPSEC or PPTP VPN
#
# NOTE #2: the $LOGGING variable is NOT included here because if it was
#          enabled, your logs would grow too quickly to manage.
#
# Ports:   137 TCP/UDP (NetBIOS name service)
#          138 UDP     (NetBIOS datagram service) - TCP filtered just in case
#          139 TCP     (NetBIOS session service)  - UDP filtered just in case
#          445 TCP/UDP (MS CIFS in Win2k)

echo "     - Rejecting TCP/UDP SMB traffic on the external interface."
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 137
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 137
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 138
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 138
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 139
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 139
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 445
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 445
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 137 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 137 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 138 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 138 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 139 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 139 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 445 -d $UNIVERSE
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 445 -d $UNIVERSE

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 137
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 137
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 138
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 138
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 139
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 139
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 445
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 445
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 137 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 137 -d $UNIVERSE
```

```
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 138 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 138 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 139 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 139 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 445 -d $UNIVERSE
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 445 -d $UNIVERSE
fi

# Explictly filter out any OUTGOING traffic that is either known to be INSECURE or from a
# possible INTERNAL machine infected with a Trojan.
#


# RPC - Used for NFS and other insecure mechanisms
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE sunrpc $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP sunrpc -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE sunrpc $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP sunrpc -d $UNIVERSE $LOGGING
fi

# Mountd - Used for NFS
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 635 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 635 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 635 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 635 -d $UNIVERSE $LOGGING
fi

# PPTP - Block unauthorized outgoing VPNs
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 1723 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 1723 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then

  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 1723 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 1723 $LOGGING
fi
```

```
# Remote Winsock - Block internal Windows machines doing weird stuff.
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 1745 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 1745 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 1745 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 1745 $LOGGING
fi


# NFS - Block NFS due to security issues
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 2049 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 2049 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 2049 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 2049 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 2049 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 2049 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 2049 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 2049 -d $UNIVERSE $LOGGING
fi

# PcAnywhere - Block unauthorized outgoing remote control sessions
#
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 5631 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 5631 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 5632 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 5632 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 5631 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 5631 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 5632 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 5632 $LOGGING
fi

# Xwindows - Block unauthorized and non-secured Xwindows
#
# NOTE: See variable section above for the example range (6000:6007 by default)
# Xwindows can use far more than just ports 6000-6007.
#
```

```
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE $XWINDOWS_PORTS $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE $XWINDOWS_PORTS $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE $XWINDOWS_PORTS $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE $XWINDOWS_PORTS $LOGGING
fi

# IPSec VPNs - Block unauthorized VPNs
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 500 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 500 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 500 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 500 $LOGGING
fi

# MySQL - Block unauthorized SQL sessions
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 3306 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 3306 -d $UNIVERSE $LOGGING
fi

# EggDrop IRC bot - Block unauthorized bots
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 3456 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 3456 -d $UNIVERSE $LOGGING
fi

# Block the following known Trojan network ports.
#
# Please note that TCP/IP, by nature uses RANDOM high ports.  So just because you get a firewall hit on
# a known trojan port doesn't always mean you have an infected internal machine.  Please also note that
# since the port in question is blocked, the local or internal IP stack will eventually use a different
# high port before giving up so things SHOULD work ok anyway.
#
# By NO means is this a complete list but I try to get the common ones.
# If I filtered out ALL the various known trojan ports, there wouldn't be many VALID high ports left!  :-(
#
#   Please see http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html for a more complete list.
```

```
#

# NetBus.
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 12345 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 12346 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 12345 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 12346 $LOGGING
fi


# NetBus Pro.
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 20034 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 20034 $LOGGING
fi

# BackOrofice
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 31337 $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP -d $UNIVERSE 31338 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 31337 $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP -d $UNIVERSE 31338 $LOGGING
fi

# Win Crash Trojan.
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 5742 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 5742 $LOGGING
fi

# Socket De Troye.
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 30303 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 30303 $LOGGING
```

```
fi

# Unknown Trojan Horse (Master's Paradise [CHR])
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP -d $UNIVERSE 40421 $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP -d $UNIVERSE 40421 $LOGGING
fi

# Trinoo UDP flooder - Please note this port will probably change over time
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 27665 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 27444 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 31335 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 27665 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 27444 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 31335 -d $UNIVERSE $LOGGING
fi


# Shaft distributed flooder - Please note this port will probably change over time
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 20432 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 18753 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 20433 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 20432 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 18753 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 20433 -d $UNIVERSE $LOGGING
fi


# SubSeven Trojan - Please note this port will probably change over time
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 27374 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p udp -s $EXTIP 27374 -d $UNIVERSE $LOGGING
/sbin/ipchains -A output -j REJECT -i $EXTIF -p tcp -s $EXTIP 1243 -d $UNIVERSE $LOGGING

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 27374 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p udp -s $INT2IP 27374 -d $UNIVERSE $LOGGING
  /sbin/ipchains -A output -j REJECT -i $INT2IF -p tcp -s $INT2IP 1243 -d $UNIVERSE $LOGGING
```

```
fi

#----------------------------------------------------------------------
# Allow all High Ports for return traffic.
#
# Some day this rule set will be stateful and we won't have to do this
#
echo "  - Enabling all output REPLY [TCP/UDP] traffic on high ports."
/sbin/ipchains -A output -j ACCEPT -p tcp -s $EXTIP $UNPRIVPORTS -d $UNIVERSE
/sbin/ipchains -A output -j ACCEPT -p udp -s $EXTIP $UNPRIVPORTS -d $UNIVERSE

# If we don't have a DMZ interface, dont do things for it
#
if [ "$INT2IF" != "" ]; then
  /sbin/ipchains -A output -j ACCEPT -p tcp -s $INT2IP $UNPRIVPORTS -d $UNIVERSE
  /sbin/ipchains -A output -j ACCEPT -p udp -s $INT2IP $UNPRIVPORTS -d $UNIVERSE
fi

#----------------------------------------------------------------------
# Catch All Rule
#----------------------------------------------------------------------
echo "  - Final output catch all rule."

# All other outgoing is denied and logged.  This rule set should catch
# everything (including samba) that hasn't already been blocked.
#
/sbin/ipchains -A output -j REJECT -s $UNIVERSE -d $UNIVERSE $LOGGING


#**********************************************************************
# Forwarding Rules
#**********************************************************************
#
echo "----------------------------------------------------------------------"
echo "Forwarding Rules:"


# Dont run these commands if MASQ isnt compiled into the kernel
if [ -a /proc/sys/net/ipv4/ip_always_defrag ] && [ "$INTIF" != "" ]; then
  #----------------------------------------------------------------------
  # Enable TCP/IP forwarding and masquerading from the Internal LAN
  #----------------------------------------------------------------------

  # Diald Users:
  #
  #  You need this rule to allow the sl0 SLIP interface to receive
  #  traffic to then bring the interface up.
  #
  #      Disabled by default
  #
```

```
#/sbin/ipchains -A forward -j MASQ -i sl0 -s $INTLAN -d $UNIVERSE


#-----------------------------------------------------------------------
# Port Forwarding
#-----------------------------------------------------------------------
# Port forwarding allows external traffic to directly connect to an INTERNAL
# Masq'ed machine. An example for this is when a user needs to have external
# users directly contact a WWW server behind the MASQ server.
#
# To use PORTFW, you need to un-# out and edit the $SECUREHOST section at
# the top of the rule set.
#
# NOTE: Port forwarding is well beyond the scope of this documentation to
#       explain the security issues implied in opening up access like this.
#       Please see Appendix A to read the IP-MASQ-HOWTO for a full explanation.
#
# Do not use ports greater than 1023 for redirection ports.
#
# Disabled by default.
#-----------------------------------------------------------------------
#echo "  * Enabling Port Forwarding onto internal hosts."
#/usr/sbin/ipmasqadm portfw -f
#echo "  * Forwarding SSH traffic on port 26 to $PORTFWIP1"
#/usr/sbin/ipmasqadm portfw -a -P tcp -L $EXTIP 26 -R $PORTFWIP1 22
#
#echo "  * Forwarding SSH traffic on port 26 to $PORTFWIP2"
#/usr/sbin/ipmasqadm portfw -a -P tcp -L $EXTIP 26 -R $PORTFWIP2 22
#
#echo "  * Forwarding SSH traffic on port 26 to $PORTFWIP3"
#/usr/sbin/ipmasqadm portfw -a -P tcp -L $EXTIP 26 -R $PORTFWIP3 22


#-----------------------------------------------------------------------
# Enable TCP/IP forwarding and masquerading from the Internal LAN
#-----------------------------------------------------------------------

# Turn on IP Forwarding in the Linux kernel
#
# There are TWO methods of turning on this feature.  The first method is the
# Red Hat way. Edit the /etc/sysconfig/network file and change the
# "FORWARD_IPV4" line to say:
#
#       FORWARD_IPV4=true
#
# The second method is shown below and can executed at any time while the
# system is running.
#
echo "  - Enabling IP forwarding."
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
  # Masquerade from local net on local interface to anywhere.
  #
  echo "  - Enable IP Masquerading from the internal LAN."
  /sbin/ipchains -A forward -j MASQ -i $EXTIF -s $INTLAN -d $UNIVERSE

  # If we don't have a DMZ interface, dont do things for it
  #
  if [ "$INT2IF" != "" ]; then
    /sbin/ipchains -A forward -j MASQ -i $EXTIF -s $INT2LAN -d $UNIVERSE
    /sbin/ipchains -A forward -j ACCEPT -i $INTIF -s $INT2LAN -d $INTLAN
    /sbin/ipchains -A forward -j ACCEPT -i $INT2IF -s $INTLAN -d $INT2LAN
  fi


  # Enabling Always Defrag for Masqueraded systems
  #
  #  Some 2.2.x and ALL 2.4.x kernels dont support this feature.
  #  If your kernel gives you an error on this line, you can safely
  #  ignore it.
  #
  echo "  - Enable IP Always Defrag for the internal LAN."
  echo "1" > /proc/sys/net/ipv4/ip_always_defrag


  # Disabling the LooseUDP patch required by some Internet-based games
  #
  # NOTE:  Some distros such as TurboLinux delete this option from the kernel
  #
  # Enabled by default
  echo "  - Disable LooseUDP [needed by some games] due to security"
  echo "1" > /proc/sys/net/ipv4/ip_masq_udp_dloose

fi

# Catch all rule, all other forwarding is denied.
#
/sbin/ipchains -A forward -j REJECT -s $UNIVERSE -d $UNIVERSE $LOGGING

#*******************************************************************
# The end
#*******************************************************************
echo "--------------------------------------------------------------------"
echo -e "TrinityOS IPCHAINS Firewall $FWVER implemented.\n\n"
#/usr/local/sbin/beep
#/usr/local/sbin/success
sleep 1
#/usr/local/sbin/beep
sleep 1
```

```
#/usr/local/sbin/beep
sleep 1
```

<TrinityOS rule set STOP>

10.8 The /etc/rc.d/init.d script to load the IPCHAINS rule set upon boot

Have the firewall rule set automatically load:

- **** IMPORTANT**** It should be noted that Mandrake 7.0+ and most likely newer Redhat versions have a section in **/etc/rc.d/rc.sysinit** to automatically load a /etc/rc.d/rc.firewall script if it exists. Since the network interfaces aren't up yet, I recommend to edit it and # out those lines

Various Linux Distributions:

- Redhat: Create the file called /etc/rc.d/init.d/firewall
- Turbo Linux: Create the /etc/rc.d/init.d/firewall file but make the following changes:
  - Change the line "chkconfig: 2345 11 89" to "chkconfig: 2345 09 91"
  - Remove the stock /etc/rc.d/init.d/ipchains script

```
--

#!/bin/sh
#
# firewall      Bring up/down networking
#
# chkconfig: 2345 11 89
# description: Loads a modified version of the TrinityOS rc.firewall rule set
# probe: true

# ----------------------------------------------------------------------------
# # TrinityOS-firewall
# v11/11/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/~dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates
# -------
#
# 11/11/00 - Fixed an echo typo to say that the policy is REJECT
#            and added a MASQ list "mlist" option
# 10/08/00 - Changed the defaults when the firewall is stopped from ACCEPT
#            to REJECT
#
```

```
# ----------------------------------------------------------------------------


# Source function library.
. /etc/rc.d/init.d/functions

# Check that networking is up.

# This line no longer work with bash2
#[ ${NETWORKING} = "no" ] && exit 0
# This should be OK.
[ "XXXX${NETWORKING}" = "XXXXno" ] && exit 0

[ -x /sbin/ifconfig ] || exit 0

# See how we were called.
case "$1" in
  start)
    /etc/rc.d/rc.firewall
    ;;
  stop)
    echo -e "\nFlushing firewall and setting default policies to REJECT\n"
    /sbin/ipchains -P input REJECT
    /sbin/ipchains -P output REJECT
    /sbin/ipchains -P forward REJECT

    /sbin/ipchains -F input
    /sbin/ipchains -F output
    /sbin/ipchains -F forward
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  status)
    /sbin/ipchains -L
    ;;
  mlist)
    /sbin/ipchains -M -L
    ;;
  *)
        echo "Usage: firewall {start|stop|restart|status|mlist}"
        exit 1
esac

exit 0

--
```

Next, make it executable:

```
                chmod 700 /etc/rc.d/init.d/firewall
```

Lastly, enable the firewall to start automatically:

```
                chkconfig --add firewall
                chkconfig --level 345 firewall on
```

Slackware:

Next, append this to the end of the "/etc/rc.d/rc.local" file

```
                #Run the IP MASQ and firewall script
                /etc/rc.d/rc.firewall
```

- Make the rc.firewall file executable

```
        chmod 700 /etc/rc.d/rc.firewall
```

Now, if you aren't running a 2.0.x kernel, please skip down to the 10.12 (Firewall Confirm) subsection to see how to safely make changes to your live firewall configuration.

```
+-------------------------------------------------------------------------------+
| rc.firewall for MASQ setups with a STRONG IPFWADM rule set for 2.0.x kernels |
|                                                                               |
|  *** Discontinued!!!  Patch your 2.0.x kernel and use the IPCHAINS rules!!   |
+-------------------------------------------------------------------------------+

/etc/rc.d/rc.firewall
```

10.9 An older TrinityOS rc.firewall rule set for 2.0.x kernels (LEGACY)

```
--
#!/bin/sh

#---------------------------------------------------------------------
# Version v2.97
#
#       NOTE to ALL IPFWADM users:
#
#               As you all know, IPFWADM has been replaced by IPCHAINS for some time
#               now.  I've also been updating the IPCHAINS rule sets for a while yet
#               the IPFWADM rule sets haven't been updated.
#
#               Though this sucks that I have to do this, I can't maintain both.
#               In the future, I will REMOVE these rule sets though I will make them
#               available via a different URL.
#
#               ** BUT... there is a kernel patch to get IPCHAINS running on 2.0.x
#               kernels.  Please see <ref id="sect-5" name="Section 5"> for the URL and use IPCHAINS from
#               now on.  Ok?
```

```
#
# v2.97 - Deleted the DHCPcd commands as the syntax was old an misleading.  Update
#         to IPCHAINS.
#
# v2.96 - Added blurbs and scripts in the EXTIP, EXTBROAD, and DGW variable areas that
#           DHCP users should use "dhcpcd" with the -c option to re-run
#           the rule set upon lease renews.  It is also mentioned that both
#           DHCP and PPP users need to get their EXTBROAD and DGW addresses
#           dynamically.
#         - Changed the debug system to re-create the debug log each time
#               (removed one of the >'s at the top of the debug setup)
#
# v2.95 - Added a /0 to the final OUTPUT reject rule.  It was implicitly there but its good
#               for documentation reasons.  There were also a few IMPUT rules that DENYed
#               instead of REJECTed traffic for spoofed traffic, etc.  Fixed.
#               I also noted that the automatic $extbroad varible will only be properly set if
#               you have a typical 255.255.255.0 netmask.  If you don't, you'll have to statically
#               define it vs. use the automatic method.
# v2.94 - Added explicit INPUT filters for NFS and OUTPUT filters for Mountd and RPC
# v2.93 - Added explicit OUTPUT filters for the BackOrofice and NetBus Windows trojans
# v2.92 - Moved the default policy settings and INPUT/OUTPUT/FORWARD flush from
#               the top of each section to the top top of the entire rule set.# v2.91
# v2.91 - Added more firewall DENY rules to stop Xwindows ports 6001-6007
# v2.90 - Changed the default policies from DENY to REJECT.
# v2.80 - Clarified the input/output rules for HTTP to use the -W interface option and
#               added a #ed out rule for allowing HTTP traffic directly to the Linux box
#               from the Internet.
# v2.75 - Added and commented on the enabling of multicast traffic
#         - Caught a serious typo:  -V CANNOT have a subnet mask appended to it.  Though
#               this is inconsitant with the other commands, this has been confirmed.
# v2.71 - Redirectted the rc.firewall debugging info to /tmp/rc.firewall.dump
# v2.70 - Added commented out debugging echo statements right after the environment vars
# v2.65 - Removed the /32 bit subnet mask from the intip, extip, dgw, secondarydns,
#               and securehost variables and manually placed them back within the rule sets
#               themselves.  This is for users who use DHCP and/or PPP that wouldn't get the
#               correct netmask.  Also, the netmask built into these variables would break
#               the IPPORTFW section.
#         - Added the LOOPBACK variable for better readibilty
#         - Cleaned the comment sections a little
#
# v2.60 - Added #'ed out rules to support the Linux box getting addressed via DHCP
# v2.51 - Corrected the vars passed to PPPd as shown bellow in the comments section
# v2.50 - Deleted an already #ed out line to allow in ALL incoming
#               traffic.
#         - Added a /32 bit subnet mask to the intip, extip, dgw, secondarydns,
#               and securehost variables.  Because of this, I then deleted a few stray
#               and possibly incorrect /24 and /32 bit masks on various IPFWADM rules
#         - Cleaned up (split up) the explicit INPUT section for internal and external
#               hosts.
#         - Cleaned up the IPPORTFW area to use all environment vars and added the
```

```
#               $portfwip var.
#         - Deleted a duplicate line for the "outgoing from local net on remote interface,
#               stuffed masquerading, deny" rule set
#
# v2.45 - Added the environment variables that PPPd passes to ease the
#               use of IPFWADM firewalls
# v2.40 - Change the default behavior of IPORTFW to disabled
#         - Made some clarifications for dynamically addressed users and
#           the "extif" variable.
# v2.30 - Commented and changed the unrestricted ports to 1024-65535
#               since SSH sometimes creates connections at port 1023
#         - Added #'ed out IPFWADM statements to do non-logged filtering
#               of BOOTP (ports 67-68), Samba (ports 137-138), RIP
#               (port 520), and SNMP (port 161)
#         - Added TCP support for DHCP
# v2.25 - Rearranged the ordering and description of the IPFWADM enviro variables
#       - Added #'ed out IPFWADM statements for WWW access to the world
# v2.20 - Addition of IPPORTFW commands
# v2.10 - Disabled ALL outbound Xwindows (Xwin uses port 6000) which was
#           previously allowed since its in the >1024 port range.  Gotcha!
# v2.00 - Totally re-written and MUCH stronger
# v1.00 - Oringial draft
#-------------------------------------------------------------------

# ++ Best viewed in a window at 90+ columns
#
# This script was adapted from Ambrose's IPMASQ-HOWTO and several
# other resources including:
#
#       - Me
#
# **Note**:  This config ASSUMES:
#
#               1) that you have your private LAN addressing set as
#                  192.168.0.x
#               2) Your internal LAN is on eth1
#               3) Your external LAN is on eth0
#               3) Your static IP address is 100.200.0.212
#                        * If you get your external IP address via DHCP, you
#                          will need to un-comment (un-#) the "DHCP - Client" rule set
#
#       Obviously, this config won't be totally correct for your
#       environment nor can your static IP address be the same
#       as mine!  So, you might need to change the IP addresses,
#         internal/external interface names, un-comment out the #'ed out DHCP client
#         lines, etc.
#
#       ---------------------------------------------------------------
#
#       This config also handles both IP spoofing and stuffed routing
```

```
#       and IP Masquerading.  Anything not explicitly allowed is
#       REJECTED.  Rejecting traffic is better than DENYING it since
#       it makes the IPFWADM'ED machine look like its not CAPABLE of
#       doing that particular protocol!
#
#       ***PPP and DHCP USERS***
#
#       1)      All PPP and DHCP users that get Dynamic IP address should
#               # out the "extip" variable a page or so down and then un-# out the
#               following command for your dynamic IP address:
#
#               NOTE: DHCP users will need to replace the "ppp0" interface name with
#                       the interface name of your external Internet interface.
#
# extip=`/sbin/ifconfig | grep -A 4 ppp0 | awk '/inet/ { print $2 } ' | sed -e s/addr://`
#
#
#       2.      Create the /etc/ppp/ip-up script file to execute this rule set:
#
#               /etc/ppp/ip-up
#               --
#               #!/bin/sh
#               /etc/rc.d/rc.firewall
#               --
#
#               NOTE:  When PPPd runs the /etc/ppp/ip-up script, it passes several
#                       environment variables which can help bring up the script.
#                       Though I haven't updated my doc to use these variables, I will
#                       at a future date:
#
#                               $1 = Interface being brought up (e.g. ppp0)
#                               $2 = TTY device being used (/dev/modem)
#                               $3 = Terminal speed (38400)
#                               $4 = IP address of my local PPP interface
#                               $5 = IP address of the remote P-t-P link (default gw)
#                               $6 = This is the IPPARM string that is passed from the
#                                       options file for any ip-up specific use
#
#
#       3.      Now make this new script executable by running "chmod 700 /etc/ppp/ip-up"

#---------------------------------------------------------------------------
#Enviroment Variables - Change to suit your environment
#

#Specification of the LOOPBACK interface
loopback="127.0.0.1"

#Specification of the INTERNAL NIC
intif="eth1"
```

```
#The IP address on your INTERNAL nic
intip="192.168.0.1"

#IP network address of the INTERNAL net
intnet="192.168.0.0"

#IP address of an internal host that should have IPPORTFW forward traffic to
portfwip="192.168.0.20"


#Specification of the EXTERNAL NIC
#
#       PPP Users: If you are using the Dynamic PPP "extif" script from above,
#               make sure to comment the below line out so it doesn’t override it.
#
#               If you want to use the PPPd variables, change this to read:
#
#               extip=ppp0
#
extif="eth0"

#The IP address you get from the Internet
#
#       PPP users: If you are getting dynamic address, either use the "extip" script
#                       from the header above or if you want to use the PPPd variables,
#                       change this to read:
#
#       EXTIP=‘/sbin/ifconfig | grep -A 4 $EXTIF | awk ’/inet/ { print $2 } ’ | sed -e s/addr://‘
#
#    NOTE:  DHCP users should also update the script that runs DHCP to
#               use "dhcpcd" instead of other solutions like RH6’s
#               "pump" DHCP solution and also have dhcpcd load.
#       It should be noted that newer versions of pump can run scripts
#       upon lease bringup, renew, etc.
#
#           This will let the firewall re-run upon DHCP lease renews
#           just in case you get a different IP address.
#
extip="100.200.0.212"


#The IP broadcast address of the external net
#
#       PPP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "extbroad" to read (this make an assuption but it should
#                       be a safe assumption):
#                       extbroad=‘echo $4 | cut -d ’.’ -f 1-3‘.255
#
#               NOTE:  This method will only work for typical 255.255.255.0 netmasks,
```

```
#                        if you get other masks such as a 255.255.252.0, you will have to
#                        statically define it like it is now instead of using the dynamic
#                        setup.
#
extbroad="100.200.0.255"

#IP address of the default gateway on the EXTERNAL NIC
#
#       PPP and DHCP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "dgw" to read:
#
#                       dgw=‘/sbin/ifconfig | grep -A 4 ppp0 | awk ’/gateway/ { print $2 } ’ | sed -e s/addr://‘
#
dgw="100.200.0.1"

#IP Mask for ALL IP addresses
universe="0.0.0.0"

#IP Mask for BROADCAST
broadcast="255.255.255.255"

#Specification of HIGH IP ports
#       NOTE: Notice that this STARTS at 1024 and NOT at 1023 which it should.
#                 for some reason SSH sometimes initiates connections at 1023 which
#                 is a TCP violation but shit happens.
#
#   Brief update:  This is due to SSH not being executed with "-P"
#
unprivports="1024:65535"

#Specification of backup DNS server
secondarydns="102.200.0.25"

#Specifically allowed external host - secure1.host.com
securehost="200.211.0.40"

#---------------------------------------------------------------------------
# Debugging Section:  If you are having problems with the firewall, uncomment
#                               out (un # out) the follow echo lines and then re-run
#                               the firewall to make sure that the rc.firewall is
#                               getting the right info.
#

#echo Loopback IP:                              $loopback >> /tmp/rc.firewall.dump
#echo Internal interface name:          $intif >> /tmp/rc.firewall.dump
#echo Internal interface IP:                    $intip >> /tmp/rc.firewall.dump
#echo Internal interface net:                   $intnet >> /tmp/rc.firewall.dump
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo External interface name:          $extif >> /tmp/rc.firewall.dump
#echo External interface IP:                    $extip >> /tmp/rc.firewall.dump
```

```
#echo External interface broadcast IP:  $extbroad >> /tmp/rc.firewall.dump
#echo External interface default gateway:       $dgw >> /tmp/rc.firewall.dump
#echo Internet IP to be port forwarded to:      $portfwip >> /tmp/rc.firewall.dump
#echo --------------------------------------------------- >> /tmp/rc.firewall.dump
#echo External secondary DNS (optional):        $secondarydns >> /tmp/rc.firewall.dump
#echo External secured host (optional): $securehost >> /tmp/rc.firewall.dump

#-----------------------------------------------------------------------------


# For a nice display
echo "  "

#Multicast is a powerful, yet seldom used aspect of TCP/IP for multimedia
#       data.  Though it isn't used much now (because most ISPs don't enable
#       multicast on their networks, it will be very common in a few more
#       years.  Check out www.mbone.com for more detail.
#
#       NOTE:  Adding this feature is OPTIONAL
#

echo "Adding multicast route.."
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev $extif

echo "Enabling IP Masquerading.."
echo "1" > /proc/sys/net/ipv4/ip_forward

#-----------------------------------------------------------------------------
# Masq timeouts
# -------------
#
# Set timeout values for masq sessions (seconds).
# I only did this because my telnet connections would drop after inactivity
# of 15 mins.

echo "Changing IP MASQ Timeouts.."
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  60 sec timeout for UDP traffic (MASQ'ed ICQ users must enable a 30sec
#                                               firewall timeout in ICQ itself)

/sbin/ipfwadm -M -s 7200 10 60

#-----------------------------------------------------------------------------

#-----------------------------------------------------------------------------
# Masq Modules
# -------------
#
echo "Loading MASQ modules.."
```

```
#/sbin/modprobe ip_masq_cuseeme
/sbin/modprobe ip_masq_ftp
#/sbin/modprobe ip_masq_irc
#/sbin/modprobe ip_masq_quake
#/sbin/modprobe ip_masq_vdolive
#/sbin/modprobe ip_masq_raudio

#---------------------------------------------------------------------------

#Set all default policies to REJECT and flush all old rules:
echo "Set all default policies to REJECT and flush all old rules"

#Change default policies
/sbin/ipfwadm -I -p reject
/sbin/ipfwadm -O -p reject
/sbin/ipfwadm -F -p reject

#Flush all old rule sets
/sbin/ipfwadm -I -f
/sbin/ipfwadm -O -f
/sbin/ipfwadm -F -f

#---------------------------------------------------------------------------
        echo "Enabling general INPUT on the internal LAN.. line 74"
#---------------------------------------------------------------------------
# INCOMING traffic on the INTERNAL LAN network
# -------------------------------------------

# local interface, local machines, going anywhere is valid
/sbin/ipfwadm -I -a accept -V $intip -S $intnet/24 -D $universe/0

# remote interface, claiming to be local machines, IP spoofing, get lost & log
/sbin/ipfwadm -I -a reject -V $extip -S $intnet/24 -D $universe/0 -o

# loopback interface is valid.
/sbin/ipfwadm -I -a accept -V $loopback -S $universe/0 -D $universe/0

# DHCP - SERVER - to serve out DHCP addresses on the internal LAN  67=bootps 68=bootpc
/sbin/ipfwadm -I -a accept -W $intif -P udp -S $universe/0 bootpc -D $broadcast/0 bootps
/sbin/ipfwadm -I -a accept -W $intif -P tcp -S $universe/0 bootpc -D $broadcast/0 bootps

## DHCP - CLIENT - if you get a dynamic IP address for your ADSL or Cablemodem connection
#/sbin/ipfwadm -I -a accept -W $extif -P udp -S $universe/0 bootps -D $broadcast/0 bootpc
#/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $universe/0 bootps -D $broadcast/0 bootpc

echo "Enabling general INPUT on the external LAN.. line 94"
#---------------------------------------------------------------------------
# INCOMING traffic on the EXTERNAL LAN network
# --------------------------------------------------------------------------
```

```
#

# Questionable... ???
# /sbin/ipfwadm -I -a accept -V $extip -P -k -S $universe/0 -D $intnet/24 $unprivports

#-----------

# ICMP: Allow ICMP from the local default GW
/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $dgw/32 -D $extip/32

## ICMP: Allow ICMP from the universe but LOG it .. nice thought but unless you
##       can figure out how to ignore REPLIES.. this is too much logging!
#/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $universe/0 -D $extip/32 -o
/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $universe/0 -D $extip/32

# NTP: Allow NTP updates tcp from any host
/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $universe/0 -D $extip/32 ntp

# IDENT: Allow IDENT on ALL interfaces but disable it in /etc/inetd.conf
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $universe/0 113

# DNS Lookups & Zone transfers: Since this site is an authoritative DNS server, we must
#                               open up DNS to the public on ALL interfaces
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $universe/0 53
/sbin/ipfwadm -I -a accept -P udp -S $universe/0 -D $universe/0 53

# SMTP MAIL: Since this site is an authoritative SMTP server, allow it in on ALL
#       interfaces.
#
#       NOTE:  No specific -W interfaces are given since I want SMTP to be available
#               from ALL interfaces and not just one specific one.
#
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $extip/32 smtp


# WWW: Allow HTTP traffic.  By default, allow all HTTP traffic from the Internal
#        LAN but DISABLE it from the Internet.  If you also require HTTP access
#         from the Internet, uncomment the #ed out rule below.
#
#Internal LAN:
/sbin/ipfwadm -I -a accept -W $intif -P tcp -S $intnet/24 -D $intip/32 www
#
#Internet:
#/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $universe/0 -D $extip/32 www

# NFS
/sbin/ipfwadm -I -a reject -W $extif -P tcp -S $universe/0 -D $extip/32 2049
/sbin/ipfwadm -I -a reject -W $extif -P tcp -S $universe/0 2049 -D $extip/32
```

```
# HIGH PORTS: Enable all HIGH ports for reply tcp/udp traffic
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $extip/32 $unprivports
/sbin/ipfwadm -I -a accept -P udp -S $universe/0 -D $extip/32 $unprivports


echo "Enabling explicit INPUT on the -INTERNAL- LAN.. line 136"
################################################################################
# Begin Explict IP INPUT allows on the INTERNAL LAN network:
################################################################################
#

### NOTE:  copy a set of the following (3) lines and modify them to reflect any
#               additional internal hosts you want to be able to access your Linux
#               box.  These examples allow FTP, FTP-DATA, SSH, and Samba.
#
#               If you want to enable TELNET access, just append the word "telnet" after
#               the word "ssh"


#coyote
/sbin/ipfwadm -I -a accept -W $intif -P tcp -S 192.168.0.2/32 -D $intip/32 ftp ftp-data ssh
/sbin/ipfwadm -I -a accept -W $intif -P udp -S 192.168.0.2/32 -D $intip/32 137 138 139

#spare
/sbin/ipfwadm -I -a accept -W $intif -P tcp -S 192.168.0.9/32 -D $intip/32 ftp ftp-data ssh
/sbin/ipfwadm -I -a accept -W $intif -P udp -S 192.168.0.9/32 -D $intip/32 137 138 139

#spare2
/sbin/ipfwadm -I -a accept -W $intif -P tcp -S 192.168.0.10/32 -D $intip/32 ftp ftp-data ssh
/sbin/ipfwadm -I -a accept -W $intif -P udp -S 192.168.0.10/32 -D $intip/32 137 138 139



echo "Enabling explicit INPUT on the -EXTERNAL- LAN.. line 136"
################################################################################
# Begin Explicit IP INPUT allows on the EXTERNAL LAN network:
################################################################################
#

### NOTE:       If you need to need to have more than just one remote Secure Host
#               into your Linux box, copy the set of (2) lines below and modify
#               them to reflect their proper IP addresses. This example allows
#               SSH and POP3 in.  In addition to this "Explict IP INPUT" exception,
#               you will need to explicitly allow this remote secure
#               host traffic to be let -OUT- of the firewall.  See the "Explict IP
#               OUTPUT allows" later in this rule set to complete the firewall rule set.
#
### NOTE2:      If you want to enable TELNET access in addition to SSH and POP3, just
#               append the word "telnet" after the word "pop-3"
#
```

```
### NOTE3:  If you want to forward FTP traffic, you will need to install a different
#               ip_masq_ftp module.  Please see the IP-MASQ-HOWTO for full details.

#secure1.host.com
/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $securehost/32 -D $extip/32 ssh pop-3



# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# IPPORTFW Re-directions..
# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#
# Port forwarding allows people from the outside to directly connect to a machine
#       on the MASQed side.  An example of this is the need for people to directly
#       contact an FTP server on  the MASQed network from the Internet.

# NOTE:  Do *NOT* use ports greater than 1023 for redirection ports.
#
#               I used to use ports 2312 for TELNET redirection but I figured out
#               that with ports > 1023, all my IPFWADM rule sets were being
#               ignored and all Internet hosts could hit my re-directed server!
#
#               Why?  Due to the default behavior of TCP/IP and MASQing, you
#               have to allow all ports > 1023 through the firewall.

##### NOTE:  Un-#ed out these statements if you want to enable IPPORTFW

#echo "Enabling IPPORTFW Redirection on the external LAN.. line 229"

#/usr/local/sbin/ipportfw -C
#/usr/local/sbin/ipportfw -A -t$extip/2112 -R $portfwip/21
#/usr/local/sbin/ipportfw -A -t$extip/2312 -R $portfwip/23
#/usr/local/sbin/ipportfw -A -t$extip/8012 -R $portfwip/80

# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# END IPPORTFW Re-directions..
# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


#  ***************************************************************************
#  ** Uncomment these non-logging IPFWADM rules if they apply to your enivroment **
#  ***************************************************************************

# Reject all stray BOOTP traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 68

# Reject all stray Samba traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $universe/0 137 138 139

# Reject all stray RIP traffic but DON'T log it since it fills up the logs fast
```

```
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $universe/0 520

# Reject all stray SNMP traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $broadcast/0 161


# Final INPUT Rule
#
# catch all rule, all other incoming is denied and logged. pity there is no
# log option on the policy but this does the job instead.
/sbin/ipfwadm -I -a reject -S $universe/0 -D $universe/0 -o



echo "Enabling general OUTPUT on the internal LAN.. line 174 "
#----------------------------------------------------------------------------
# OUTGOING traffic on the INTERNAL LAN network
# --------------------------------------------

# local interface, any source going to local net is valid
/sbin/ipfwadm -O -a accept -V $intip -S $universe/0 -D $intnet/24

# outgoing to local net on remote interface, stuffed routing, deny & log
/sbin/ipfwadm -O -a reject -V $extip -S $universe/0 -D $intnet/24 -o

# outgoing from local net on remote interface, stuffed masquerading, deny
/sbin/ipfwadm -O -a reject -V $extip -S $intnet/24 -D $universe/0 -o

#DISABLED - Too open
## anything else outgoing on remote interface is valid
#ipfwadm -O -a accept -V $extip -S $extip/32 -D $universe/0

# loopback interface is valid.
/sbin/ipfwadm -O -a accept -V $loopback -S $universe/0 -D $universe/0

# DHCP - SERVER - to serve out DHCP addresses on the internal LAN  67=bootps 68=bootpc
/sbin/ipfwadm -O -a accept -W $intif -P udp -S $intip/32 bootps -D $broadcast/0 bootpc
/sbin/ipfwadm -O -a accept -W $intif -P tcp -S $intip/32 bootps -D $broadcast/0 bootpc

## DHCP - CLIENT - if you get a dynamic IP address for your ADSL or Cablemodem connection
#/sbin/ipfwadm -O -a accept -W $extif -P udp -S $universe/0 bootpc -D $broadcast/0 bootps
#/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $universe/0 bootpc -D $broadcast/0 bootps


echo "Enabling general OUTPUT on the EXTERNAL LAN.. line 204 "
#----------------------------------------------------------------------------
# OUTGOING traffic on the external LAN network
# --------------------------------------------
# ICMP:  Allow ICMP traffic out
/sbin/ipfwadm -O -a accept -P icmp -S $universe/0 -D $universe/0
```

```
# NTP: Allow NTP updates tcp from any host
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 ntp -D $universe/0

# IDENT: Allow IDENT out but have it disabled in /etc/inetd.conf
/sbin/ipfwadm -O -a accept -P tcp -S $universe/0 113 -D $universe/0

# DNS Lookups & Zone transfers: Since this site is an authoritative DNS
#                               server, we must open up DNS to the public
#                               on ALL interfaces
#                               - You do not need port 42?
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 53 -D $universe/0
/sbin/ipfwadm -O -a accept -P udp -S $extip/32 53 -D $universe/0

# SMTP MAIL: Since this site is an authoritative SMTP server, allow it in on ALL
#       interfaces
#
#       NOTE:  No specific -W interfaces are given since I want SMTP to be available
#               from ALL interfaces and not just one specific one.
#
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 smtp -D $universe/0


# WWW:   Allow HTTP traffic.  By default, allow all HTTP traffic from the
#        Internal LAN but DISABLE it from the Internet.  If you also require
#        HTTP access from the Internet, uncomment the #ed out rule below.
#
#Internal LAN:
/sbin/ipfwadm -O -a accept -W $intif -P tcp -S $intip/32 www -D $intnet/24
#
#Internet:
#/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 www -D $universe/0

# RPC - reject
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 111 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 111 -D $universe/0 -o

# Mountd - reject
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 635 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 635 -D $universe/0 -o

# PPTP - reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 1723 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 1723 -o

# Remote Winsock - Reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 1745 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 1745 -o

# NFS - Reject
```

```
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 2049 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 2049 -D $universe/0 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 2049 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 2049 -D $universe/0 -o

# PcAnywhere - Reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 5631 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 5631 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 5632 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 5632 -o

# Xwindows - Deny
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6000 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6001 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6002 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6003 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6004 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6005 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6006 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6007 -o
#
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6000 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6001 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6002 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6003 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6004 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6005 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6006 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6007 -o

# NetBus: REJECT Netbus and LOG it
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 12345 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 12346 -o


# BackOrofice: REJECT BO on LOG it
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 31337 -o



# HIGH PORTS: Enable all HIGH ports for reply tcp/udp traffic
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 $unprivports -D $universe/0
/sbin/ipfwadm -O -a accept -P udp -S $extip/32 $unprivports -D $universe/0



echo "Enabling explicit OUTPUT on the external LAN.. line 231"
################################################################################
# Begin Explict IP OUTPUT allows on the EXTERNAL LAN network:
################################################################################
#
### NOTE:       If you need to need to have more than just one remote Secure Host
```

```
#               into your Linux box, copy the set of (2) lines below and modify
#               them to reflect their proper IP addresses. This example allows
#               FTP, FTP-DATA, SSH, and POP3 out.  In addition to this "Explict IP
#               OUTPUT" exception, you will need to explicitly allow this remote secure
#               host traffic to be let -IN- to the firewall.  See the "Explict IP
#               INPUT allows" previously in this rule set to complete the firewall
#               rule set.
#
### NOTE2:      If you want to enable TELNET access in addition to FTP, FTP-DATA,
#               and POP3, just append the word "telnet" after the word "pop-3"


#secure1.host.com
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 ftp ftp-data ssh pop-3 -D $securehost/32 $unprivports


# +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


##############################################################################
# End Explict IP OUTPUT allows:
##############################################################################

# catch all rule, all other outgoing is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
# This should catch everything including SAMBA an all non-explicitly allowed
#   TELNET, FTP, FTP-DATA, SSH, etc.
/sbin/ipfwadm -O -a reject -S $universe/0 -D $universe/0 -o


echo "Enabling MASQ on the external LAN.. line 250"
#---------------------------------------------------------------------------
# Forwarding traffic from the internal LAN network
# -------------------------------------------
#

# Masquerade from local net on local interface to anywhere.
/sbin/ipfwadm -F -a masquerade -W $extif -S $intnet/24 -D $universe/0

# catch all rule, all other forwarding is denied and logged. pity there is no
# log option on the policy but this does the job instead.
/sbin/ipfwadm -F -a reject -S $universe/0 -D $universe/0 -o

#--------------------------------------------------------------------
# For a nice display
echo "  "
--
```

Redhat:

edit /etc/rc.d/init.d/network and find where the [STAR] block ends (search for the sentance "stop") and ADD the following just above the double semi-colons ";;"

```
                        /etc/rc.d/init.d/network
                        --
                        #Run the IP MASQ and firewall script
                        /etc/rc.d/rc.firewall
                        --
```

Slackware:

Next, append this to the end of the "/etc/rc.d/rc.local" file

---

```
                --
                #Run the IP MASQ and firewall script
                /etc/rc.d/rc.firewall
```

–

- Make the rc.firewall file executable

```
        chmod 700 /etc/rc.d/rc.firewall
```

Now, if you aren't running a 2.0.x kernel for non-Masq users, please skip down to the 10.12 (Firewall Confirm) subsection to see how to safely make changes to your live firewall configuration.

```
#############################################################################
# NON-MASQ rc.firewall                                                      #
#                                                                           #
#     The follwing IPFWADM rule set, based upon the rule set above, is for  #
#     NON-MASQ users who just want to restrict access to their Linux box.   #
#     This current config allows gloabal acces to:                          #
#                                                                           #
#           - DNS, SENDMAIL, WWW                                            #
#                                                                           #
#   But it restricts access to only a few IPS for:                          #
#                                                                           #
#           - SSH, FTP, FTP-DATA, and POP-3                                 #
#############################################################################




+------------------------------------------------+
| rc.firewall for NON-MASQ setups using IPFWADM |
|                                                |
|  *** Discontinued!!!  Patch your 2.0.x kernel |
|      and use the IPCHAINS rules!!             |
+------------------------------------------------+
```

10.10 An older TrinityOS rc.firewall rule set for 2.0.x kernels not running IP-MASQ (LEGACY)

/etc/rc.d/rc.firewall

```
--
#!/bin/sh

#----------------------------------------------------------------------
# Version v2A.97
#
#       NOTE to ALL IPFWADM users:
#
#               As you all know, IPFWADM has been replaced by IPCHAINS for some time
#               now.  I've also been updating the IPCHAINS rule sets for a while yet
#               the IPFWADM rule sets haven't been updated.
#
#               Though this sucks that I have to do this, I can't maintain both.
#               In the future, I will REMOVE these rule sets though I will make them
#               available via a different URL.
#
#               ** BUT... there is a kernel patch to get IPCHAINS running on 2.0.x
#               kernels.  Please see <ref id="sect-5" name="Section 5"> for the URL and use IPCHAINS from
#               now on.  Ok?
#
# v2A.97 - Fixed a typo in the BackOrofice filter.  It was using the var
#               exitif vs. the correct extif.
#
# v2A.96 - Added blurbs and scripts in the EXTIP, EXTBROAD, and DGW variable
#               areas that DHCP users should use "dhcpcd" with the -c option to re-run
#               the rule set upon lease renews.  It is also mentioned that both
#               DHCP and PPP users need to get their EXTBROAD and DGW addresses
#               dynamically.
#
#          - Changed the debug system to re-create the debug log each time
#               (removed one of the >'s at the top of the debug setup)
#
# v2A.95 - Added a /0 to the final OUTPUT reject rule.  It was implicitly there but its good
#               for documentation reasons.  There were also a few IMPUT rules that DENYed
#               instead of REJECTed traffic for spoofed traffic, etc.  Fixed.
#               I also noted that the automatic $extbroad varible will only be properly set if
#               you have a typical 255.255.255.0 netmask.  If you don't, you'll have to statically
#               define it vs. use the automatic method.
# v2A.94 - Added explicit INPUT filters for NFS and OUTPUT filters for Mountd and RPC
# v2A.93 - Added explicit OUTPUT filters for the BackOrofice and NetBus Windows trojans
# v2A.92 - Moved the default policy settings and INPUT/OUTPUT/FORWARD flush from
#               the top of each section to the top top of the entire rule set.
# v2A.91 - Added more firewall DENY rules to stop Xwindows ports 6001-6007
# v2A.90 - Changed the default policies from DENY to REJECT.
# v2A.80 - Clarified the input/output rules for HTTP to use the -W interface
#               option.
```

```
# v2A.75 - Added and commented on the addition of multicast traffic
#          - Caught a serious typo:  -V CANNOT have a subnet mask appended to it.  Though
#               this is inconsitant with the other commands, this has been confirmed.
# v2A.71 - Redirectted the rc.firewall debugging info to /tmp/rc.firewall.dump
# v2A.70 - Added commented out debugging echo statements right after the environment vars
#          - Deleted the un-used $intif, $intip, and $intnet environment vars
#
# v2A.65 - Removed the /32 bit subnet mask from the intip, dgw, secondarydns,
#               and securehost variables and manually placed them back within the rule sets
#               themselves.  This is for users who use DHCP and/or PPP that wouldn't get the
#               correct netmask.  Also, the netmask built into these variables would break
#               the IPPORTFW section.
#          - Added the LOOPBACK variable for better readibilty
#          - Cleaned the comment sections a little
#
# v2A.60 - Added #'ed out rules to support the Linux box getting addressed via DHCP
# v2A.51 - Corrected the vars passed to PPPd as shown bellow in the comments section
# v2A.50 - Deleted an already #ed out line to allow in ALL incoming
#               traffic.
#          - Added a /32 bit subnet mask to the intip, extip, dgw, secondarydns,
#               and securehost variables.  Because of this, I then deleted a few stray
#               and possibly incorrect /24 and /32 bit masks on various IPFWADM rules
# v2A.45 - Added the environment variables that PPPd passes to ease the
#               use of IPFWADM firewalls
# v2A.40 - Made some clarifications for dynamically addressed users and
#            the "extif" variable.
# v2A.30 - Added the better commented environment vars
#          - Added #'ed out IPFWADM statements to do non-logged filtering
#               of BOOTP (ports 67-68), Samba (ports 137-138), RIP
#               (port 520), and SNMP (port 161)
#          - Deleted out all the leftover header docments that were
#               specific to the MASQ firewall
#          - Added TCP support for DHCP
#          - Fixed outgoing DNS to reflect port 53 on the SOURCE packet
#
# v2A.20 - New rev for firewalling of a single interface server
#
#-------------------------------------------------------------------

# ++ Best viewed in a window at 90+ columns
#
# This script was adapted from Ambrose's IPMASQ-HOWTO and several
# other resources including:
#
#       - Me
#
# **Note**:  This config ASSUMES:
#               1) Your external LAN is on eth0
#               2) Your static IP address is 100.200.0.212
#
```

```
#       Obviously, this config won't be totally correct for your
#       environment nor can your static IP address be the same
#       as mine!
#
#       So, you'll need to either manually change the IP address in
#       the environment variable section or or use the following
#       command to set it up for you.
#
#       This config also handles both IP spoofing and stuffed routing
#       and IP Masquerading.  Anything not explicitly allowed is
#       REJECTED.  Rejecting traffic is better than DENYING it since
#       it makes the IPFWADM'ED machine look like its not CAPABLE of
#       doing that particular protocol!
#
#       ***PPP USERS***
#
#       1)      All PPP users that get Dynamic IP address should
#               # out the "extip" variable a page or so down and then un-# out the
#               following command for your dynamic IP address:
#
# extip=`/sbin/ifconfig | grep -A 4 ppp0 | awk '/inet/ { print $2 } ' | sed -e s/addr://`
#
#       2.      Create the /etc/ppp/ip-up script file to execute this rule set:
#
#               /etc/ppp/ip-up
#               --
#               #!/bin/sh
#               /etc/rc.d/rc.firewall
#               --
#
#               Now make this new script executable by running "chmod 700 /etc/ppp/ip-up"
#
#               NOTE:  When PPPd runs the /etc/ppp/ip-up script, it passes several
#                       environment variables which can help bring up the script.
#                       Though I haven't updated my doc to use these variables, I will
#                       at a future date:
#
#                               $1 = Interface being brought up (e.g. ppp0)
#                               $2 = TTY device being used (/dev/modem)
#                               $3 = # Terminal speed (38400)
#                               $4 = IP address of my local PPP interface
#                               $5 = IP address of the remote P-t-P link (default gw)
#                               $6 = This is the IPPARM string that is passed from the options
#                                       file for any ip-up specific use
#
#       3.      Now make this new script executable by running "chmod 700 /etc/ppp/ip-up"

#-----------------------------------------------------------------------------
#Enviroment Variables - Change to suit your environment
#
```

```
#Specification of the LOOPBACK interface
loopback="127.0.0.1"

#Specification of the EXTERNAL NIC
#
#       PPP Users: If you are using the Dynamic PPP "extif" script from above,
#               make sure to comment the below line out so it doesn't override it.
#
#               If you want to use the PPPd variables, change this to read:
#                       extif="$1"
#
extif="eth0"

#The IP address you get from the Internet
#
#       PPP users: If you are getting dynamic address, either use the "extip" script
#                       from the header above or if you want to use the PPPd variables,
#                       change this to read:
#                       extip="$3"
#
#               or you can use the following script:
#
#       EXTIP=`/sbin/ifconfig | grep -A 4 $EXTIF | awk '/inet/ { print $2 } ' | sed -e s/addr://`
#
#
#    DHCP users:  DHCP users should also update the script that runs DHCP to
#                       use "dhcpcd" instead of other solutions like RH6's
#                       "pump" DHCP solution.  It should be noted that newer
#           versions of pump can run scripts upon lease bringup, renew, etc.
#           Fow now, have dhcpcd load with the option:
#
#                               -c /etc/rc.d/rc.firewall.ipchains
#
#           This will let the firewall re-run upon DHCP lease renews
#          just in case you get a different IP address.
#
extip="100.200.0.212"


#The IP broadcast address of the external net
#
#       PPP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "extbroad" to read (this make an assuption but it should
#                       be a safe assumption):
#                       extbroad=`echo $4 | cut -d '.' -f 1-3`.255
#
#               NOTE:  This method will only work for typical 255.255.255.0 netmasks,
#                        if you get other masks such as a 255.255.252.0, you will have to
#                        statically define it like it is now instead of using the dynamic
```

```
#                         setup.
#
extbroad="100.200.0.255"

#IP address of the default gateway on the EXTERNAL NIC
#
#       PPP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "dgw" to read:
#                       dgw=$4
#
#               or
#
#                       dgw=`/sbin/ifconfig | grep -A 4 ppp0 | awk '/gateway/ { print $2 } ' | sed -e s/addr://`
#
dgw="100.200.0.1"

#IP Mask for ALL IP addresses
universe="0.0.0.0"

#IP Mask for BROADCAST
broadcast="255.255.255.255"

#Specification of HIGH IP ports
#       NOTE: Notice that this STARTS at 1024 and NOT at 1023 which it should.
#                 for some reason SSH sometimes initiates connections at 1023 which
#                 is a TCP violation but shit happens.
#
#   Brief update:  This is due to SSH not being executed with "-P"
#
unprivports="1024:65535"

#Specification of backup DNS server
secondarydns="102.200.0.25"

#Specifically allowed external host - secure1.host.com
securehost="200.211.0.40"

#---------------------------------------------------------------------------
# Debugging Section:  If you are having problems with the firewall, uncomment
#                                out (un # out) the follow echo lines and then re-run
#                                the firewall to make sure that the rc.firewall is
#                                getting the right info.
#

#echo Loopback IP:                             $loopback > /tmp/rc.firewall.dump
#echo ------------------------------------------------------ >> /tmp/rc.firewall.dump
#echo External interface name:                 $extif >> /tmp/rc.firewall.dump
#echo External interface IP:                   $extip >> /tmp/rc.firewall.dump
#echo External interface broadcast IP:  $extbroad >> /tmp/rc.firewall.dump
#echo External interface default gateway:       $dgw >> /tmp/rc.firewall.dump
```

```
#echo ----------------------------------------------------- >> /tmp/rc.firewall.dump
#echo External secondary DNS (optional):        $secondarydns >> /tmp/rc.firewall.dump
#echo External secured host (optional): $securehost >> /tmp/rc.firewall.dump

#----------------------------------------------------------------------------


# For a nice display
echo "  "

#Multicast is a powerful, yet seldom used aspect of TCP/IP for multimedia
#       data.  Though it isn't used much now (because most ISPs don't enable
#       multicast on their networks, it will be very common in a few more
#       years.  Check out www.mbone.com for more detail.
#
#       NOTE:  Adding this feature is OPTIONAL
#
echo "Adding multicast route.."
/sbin/route add -net 224.0.0.0 netmask 240.0.0.0 dev $extif

#----------------------------------------------------------------------------

#Set all default policies to REJECT and flush all old rules:
echo "Set all default policies to REJECT and flush all old rules"

#Change default policies
/sbin/ipfwadm -I -p reject
/sbin/ipfwadm -O -p reject
/sbin/ipfwadm -F -p reject

#Flush all old rule sets
/sbin/ipfwadm -I -f
/sbin/ipfwadm -O -f
/sbin/ipfwadm -F -f

#----------------------------------------------------------------------------
        echo "Enabling general INPUT on the external LAN.. line 74"
#----------------------------------------------------------------------------
# INCOMING traffic on the EXTERNAL LAN network
# ------------------------------------------
#

# local interface, local machines, going anywhere is valid
#/sbin/ipfwadm -I -a accept -V $extip -S $intnet/24 -D $universe/0

# remote interface, claiming to be local machines, IP spoofing, get lost & log
#/sbin/ipfwadm -I -a reject -V $extip -S $intnet/24 -D $universe/0 -o

# loopback interface is valid.
/sbin/ipfwadm -I -a accept -V $loopback -S $universe/0 -D $universe/0
```

```
# DHCP - SERVER - to serve out DHCP addresses on the internal LAN  67=bootps 68=bootpc
#/sbin/ipfwadm -I -a accept -W $intif -P udp -S $universe/0 bootpc -D $broadcast/0 bootps

## DHCP - CLIENT - if you get a dynamic IP address for your ADSL or Cablemodem connection
#/sbin/ipfwadm -I -a accept -W $extif -P udp -S $universe/0 bootps -D $broadcast/0 bootpc
#/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $universe/0 bootps -D $broadcast/0 bootpc


# Questionable... ???
# /sbin/ipfwadm -I -a accept -V $extip -P -k -S $universe/0 -D $intnet/24 $unprivports

#-----------

# ICMP: Allow ICMP from the local default GW
/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $dgw/32 -D $extip/32

## ICMP: Allow ICMP from the universe but LOG it .. nice thought but unless you
##       can figure out how to ignore REPLIES.. this is too much logging!
#/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $universe/0 -D $extip/32 -o
/sbin/ipfwadm -I -a accept -W $extif -P icmp -S $universe/0 -D $extip/32

# NTP: Allow NTP updates tcp from any host
/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $universe/0 -D $extip/32 ntp

# IDENT: Allow IDENT on ALL interfaces but disable it in /etc/inetd.conf
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $universe/0 113

# DNS Lookups & Zone transfers: Since this site is an authoritative DNS server, we must
#                               open up DNS to the public on ALL interfaces
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $universe/0 53
/sbin/ipfwadm -I -a accept -P udp -S $universe/0 -D $universe/0 53

# SMTP MAIL: Since this site is an authoritative SMTP server, allow it in on ALL
#       interfaces
#
#       NOTE:  No specific -W interfaces are given since I want SMTP to be available
#               from ALL interfaces and not just one specific one.
#
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $extip/32 smtp

# WWW: Since this site is an authoritative WWW server, allow it in on ALL
#       interfaces
/sbin/ipfwadm -I -a accept -P tcp -W $extif -S $universe/0 -D $extip/32 www

# NFS
/sbin/ipfwadm -I -a reject -W $extif -P tcp -S $universe/0 -D $extip/32 2049
/sbin/ipfwadm -I -a reject -W $extif -P tcp -S $universe/0 2049 -D $extip/32

# HIGH PORTS: Enable all HIGH ports for reply tcp/udp traffic
```

```
/sbin/ipfwadm -I -a accept -P tcp -S $universe/0 -D $extip/32 $unprivports
/sbin/ipfwadm -I -a accept -P udp -S $universe/0 -D $extip/32 $unprivports


echo "Enabling explicit INPUT on the external LAN.. line 136"
###############################################################################
# Begin Explict IP INPUT allows on the EXTERNAL LAN network:
###############################################################################
#

#securehost
/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $securehost/32 -D $extip/32 ftp ftp-data ssh

#
###############################################################################
# End Explict IP INPUT allows on the EXTERNAL LAN network:
###############################################################################

#  ******************************************************************************
#  ** Uncomment these non-logging IPFWADM rules if they apply to your enivroment **
#  ******************************************************************************

# Reject all stray BOOTP traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 68

# Reject all stray Samba traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $universe/0 137 138 139

# Reject all stray RIP traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $universe/0 520

# Reject all stray SNMP traffic but DON'T log it since it fills up the logs fast
#/sbin/ipfwadm -I -a reject -P udp -S $universe/0 -D $broadcast/0 161


# catch all rule, all other incoming is denied and logged. pity there is no
# log option on the policy but this does the job instead.
/sbin/ipfwadm -I -a reject -S $universe/0 -D $universe/0 -o

echo "Enabling general OUTPUT on the external LAN.. line 174 "
#-----------------------------------------------------------------------------
# OUTGOING traffic on the EXTERNAL LAN network
# --------------------------------------------

# local interface, any source going to local net is valid
#/sbin/ipfwadm -O -a accept -V $intip -S $universe/0 -D $intnet/24

# outgoing to local net on remote interface, stuffed routing, deny & log
#/sbin/ipfwadm -O -a reject -V $extip -S $universe/0 -D $intnet/24 -o
```

```
# outgoing from local net on remote interface, stuffed masquerading, deny
#/sbin/ipfwadm -O -a reject -V $extip -S $intnet/24 -D $universe/0 -o

# outgoing from local net on remote interface, stuffed masquerading, deny
#/sbin/ipfwadm -O -a reject -V $extip -S $universe/0 -D $intnet/24 -o

# loopback interface is valid.
/sbin/ipfwadm -O -a accept -V $loopback -S $universe/0 -D $universe/0

# DHCP - SERVER - to serve out DHCP addresses on the internal LAN  67=bootps 68=bootpc
#/sbin/ipfwadm -O -a accept -W $intif -P udp -S $intip/32 bootps -D $broadcast/0 bootpc

## DHCP - CLIENT - if you get a dynamic IP address for your ADSL or Cablemodem connection
#/sbin/ipfwadm -O -a accept -W $extif -P udp -S $universe/0 bootpc -D $broadcast/0 bootps
#/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $universe/0 bootpc -D $broadcast/0 bootps

echo "Enabling general OUTPUT on the EXTERNAL LAN.. line 204 "

# -------------------------------------------
# ICMP:  Allow ICMP traffic out
/sbin/ipfwadm -O -a accept -P icmp -S $universe/0 -D $universe/0

# NTP: Allow NTP updatestcp from any host
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 ntp -D $universe/0

# IDENT: Allow IDENT out but have it disabled in /etc/inetd.conf
/sbin/ipfwadm -O -a accept -P tcp -S $universe/0 113 -D $universe/0

# DNS Lookups & Zone transfers: Since this site is an authoritative DNS
#                               server, we must open up DNS to the public
#                               on ALL interfaces
#                               - You do not need port 42?
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 53 -D $universe/0
/sbin/ipfwadm -O -a accept -P udp -S $extip/32 53 -D $universe/0

# SMTP MAIL: Since this site is an authoritative SMTP server, allow it in on ALL
#       interfaces
#
#       NOTE:  No specific -W interfaces are given since I want SMTP to be available
#               from ALL interfaces and not just one specific one.
#
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 smtp -D $universe/0

# WWW: Since this site is an authoritative www server, allow it in on ALL
#       interfaces
/sbin/ipfwadm -O -a accept -P tcp -W $extif -S $extip/32 www -D $universe/0

# RPC - reject
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 111 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 111 -D $universe/0 -o
```

```
# Mountd - reject
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 635 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 635 -D $universe/0 -o

# PPTP - reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 1723 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 1723 -o

# Remote Winsock - Reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 1745 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 1745 -o

# NFS - Reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 2049 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 2049 -D $universe/0 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 2049 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 2049 -D $universe/0 -o

# PcAnywhere - Reject
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 5631 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 5631 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 5632 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 5632 -o

# Xwindows - Deny
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6000 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6001 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6002 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6003 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6004 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6005 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6006 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 6007 -o
#
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6000 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6001 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6002 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6003 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6004 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6005 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6006 -o
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 6007 -o

# NetBus: REJECT Netbus and LOG it
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 12345 -o
/sbin/ipfwadm -O -a reject -W $extif -P tcp -S $extip/32 -D $universe/0 12346 -o

# BackOrofice: REJECT BO on LOG it
/sbin/ipfwadm -O -a reject -W $extif -P udp -S $extip/32 -D $universe/0 31337 -o
```

```
# HIGH PORTS: Enable all HIGH ports for reply tcp/udp traffic
/sbin/ipfwadm -O -a accept -P tcp -S $extip/32 $unprivports -D $universe/0
/sbin/ipfwadm -O -a accept -P udp -S $extip/32 $unprivports -D $universe/0


echo "Enabling explicit OUTPUT on the external LAN.. line 231"
##############################################################################
# Begin Explict IP OUTPUT allows on the EXTERNAL LAN network:
##############################################################################
#

#securehost
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 ftp ftp-data ssh -D $securehost/32 $unprivports


# ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


##############################################################################
# End Explict IP OUTPUT allows:
##############################################################################

# catch all rule, all other outgoing is denied and logged. pity there is no
# log option on the policy but this does the job instead.
#
# This should catch everything including SAMBA an all non-explicitly allowed
#   TELNET, FTP, FTP-DATA, SSH, etc.
/sbin/ipfwadm -O -a reject -S $universe/0 -D $universe/0 -o


#---------------------------------------------------------------------------
# Forwarding traffic from the internal LAN network
# --------------------------------------------

# catch all rule, all other forwarding is denied and logged. pity there is no
# log option on the policy but this does the job instead.
/sbin/ipfwadm -F -a reject -S $universe/0 -D $universe/0 -o

#--------------------------------------------------------------------
# For a nice display
echo "  "

# --end
--
```

Next, append this to the end of the "/etc/rc.d/rc.local" file

All distributions:

```
        --
```

```
        #Run the IP MASQ and firewall script
        /etc/rc.d/rc.firewall
        --
```

- Make the rc.firewall file executable

```
        chmod 700 /etc/rc.d/rc.firewall
```

10.11 Tips on editing the rc.firewall to support specific access

First, you need to figure out what kind of access you are looking for. Ideally (in the name of security), you shouldn't allow the entire Internet to acces your box but only a few IP addresses.

If you can restrict the access down to a few IPs ——————————————————— First, edit the rc.firewall ruleset that you have already modified to fit your needs and un-# out one or more of the SECUREHOST variables towards the top. Here, you will put in your desired remote IP addresses that you want to allow into your box. Next, un-# out the respective SECUREHOST lines in both the INPUT and OUTPUT sections of the rule. One critical thing to change on these two sets of lines is to change the PORT number to reflect the port you want to allow in (23 for telnet, 21 for ftp, etc). Finally, if you actually want to PORTFW this traffic to some internal machine behind a MASQ user, you will want to jump to the section below.

Setting up PORTFW

————————— To do PORTFW, you need to to towards the top of the rc.firewall file and you need to un-# a PORTFWIP variable. Here, you need to put in the IP address of the internal server you want to contact on, say port 23. Once this is done, you need to goto the PORTFW section of TrinityOS (almost at the very end) and un-# out the line for the respective PORTFW variable you just enabled. Don't forget to update the various TCP/IP ports in the PORTFW example line to be port 23 and 23 where as the example uses 26 and 22.

Thats it.. re-run the firewall and you should be good to go.

10.12 Testing your firewall rulesets

```
#----------------------------------------------------------------------
# How to test your new firewall..
#
#       From the IPFWADM console:
#
#               TELNET: telnet to a remote site
#               SSH:    ssh to a remote site
#               DNS:    run nslookup with "server = " and "set q ="
#               NTP:    run "/etc/cron.15min/gettime"
#               Xwin: "export DISPLAY=your-remote-FQDN:0.0"
#                       Run a X-server on the remote machine
#                       Run "xeyes"
#
#       From a MASQed computer on the internal LAN:
#
#       From another machine on the Internet:
#               TELNET: telnet to your IPFWADMed machine
#               SSH:    SSH to your IPFWADMed machine
```

```
#
#       ***     Finally.. download "nmap" (URL is in [Section 5] and run it
#               in both SOCKET and UDP mode to port scan your new firewall!
#
```

10.13 Remotely running the firewall-confirm file

One thing that ALL users need to be absolutely PERFECT with is making changes to their firewall rulesets **remotely**. If you were to make one ill-placed mistake, your firewall machine could become unresponsive to ALL network traffic. This means all incoming and outgoing traffic be it SMTP, WWW, even PINGs could be dropped.

To be sure that you don't take your remote machine offline, create this script file:

/usr/local/sbin/firewall-confirm

```
#!/bin/sh

# ----------------------------------------------------------------------------
# # TrinityOS-firewall-confirmed
# v11/09/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/~dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates
# -------
#
# 11/09/00 - The initial release was the wrong version.  Ack!  This updated
#            version includes a critical check for /tmp/fwok.  This version
#            also includes a 30 second screen timer.
#            Please upgrade!
#
# ----------------------------------------------------------------------------


# This script should be run when editing and running a new firewall
# version remotely.
#
# When you run this command, you will have 30 seconds to:
#
#    touch /tmp/fwok
#
# If this script doesn't see it in 30 seconds, it will revert back
# to the old firewall.


if [ ! -f /etc/rc.d/rc.firewall-checked ]; then
```

```
  echo -e "rc.firewall-checked missing.. aborting!\n\n"
  exit
fi

if [ -f /tmp/fwok ]; then
  echo -e "rc.firewall /tmp/fwok already exists.. aborting!\n\n"
  exit
fi

echo "Command Line options: $1"

echo -e "Running /etc/rc.d/rc.firewall\n\n"
/etc/rc.d/rc.firewall &


echo -e "You have 30 seconds to create /tmp/fwok..\n"

# Verbose wait loop
i=1
while [ $i -le 30 ]; do
echo -n "[$i]"
sleep 1
i=$((i=$i+1))
done
echo -e "\nWait loop complete.."


if [ ! -f /tmp/fwok ]; then
  echo -e "Rolling back to last known good config\n\n"
  /etc/rc.d/rc.firewall-checked
 else
  echo -e "\n/tmp/fwok found.. new firewall took effect..\n\n"
  rm -f /tmp/fwok
fi
```

Now, don't forget to make it executable:

```
   chmod 700 /usr/local/sbin/firewall-confirmed
```

Ok.. to use this script, do the following:

- Make a opy of a known GOOD /etc/rc.d/rc.firewall script

```
            cp /etc/rc.d/rc.firewall /etc/rc.d/rc.firewall-checked
```

- Ok.. so now go ahead and make your required changes to the /etc/rc.d/rc.firewall ruleset but DO NOT RUN IT directly.
- Ok.. when you are ready to run the new ruleset, run the following command instead:

```
          /usr/local/sbin/firewall-confirmed &
```

Please don't forget the "&" at the end to run the script in the background.

- The firewall will now load and you notice a message telling you that you have 30 seconds to create the /tmp/fwok file.
- At this point, if things are going well, you will see a counter counting up to 30. It is important that you run the command:

```
        touch /tmp/fwok
```

within those 30 seconds or the script will automatically revert back to the known good rc.firewall-checked ruleset. AaaHa! There is the beauty! If there was a critical error in your new rc.firewall ruleset, you wouldn't have ever seen that counter because your network connection would have been lost. But, because you weren't able to create that /tmp/fwok file, the firewall-confirmed script would run the the known good rc.firewall-checked file. So, in a worst-case senario, your network connection might have been disconnected but you would be still be able to re-connect to the firewall machine, fix your mistake, and try again! Cool eh?

11 Initial Preparation for Kernel Patching and Compiling

If you have a WWW server, a POP3 server, etc... (say 192.168.0.2) running behind your MASQing Linux box, you can have the MASQ box forward ALL port 80, port 110, etc connections sent to 192.168.0.2 automatically!

With the stock kernel, you CANNOT port forward FTP traffic or many non-NAT friendly Internet games properly to an internal MASQed host. To do this, you need to apply kernels patches, compile up a new IP_MASQ_FTP kernel module, etc. Though these specific topics are not covered in TrinityOS, they ARE fully covered in the new IP-MASQ-HOWTO that I have written. This new HOWTO is available on the IP MASQ WWW site and the URL for this site in in 5 (Section 5)

NOTE #2: Many people use IPAUTOFW for this function and it does work. But, I have to warn you, I have seen and PROVEN that IPAUTOFW can cause both performance and reliability issues even when compiled IN! Just don't use IPAUTOFW. Use IPPORTFW.

If you are running a 2.2.x kernel, you will need to use the new tool called IPMASQADM. Please see the IP-MASQ-HOWTO found in 5 (Section 5) for FULL details.

IPPORTFW for 2.0.x kernels allow for direct connections from the Internet to connect to one of your internal privately addressed servers. Linux 2.2.x kernels have this functionality built in.

- First, you might be concerned about security with PORTFWing, but this is what Steven had to say about that (the author of IPPORTFW):

"Port Forwarding is only called within masquerading functions so it fits inside the same ipfwadm rules. Masquerading is an extension to IP forwarding. Therefore, ipportfw only sees a packet if it fits both the input and masquerading ipfwadm rule sets."

From this and my IPFWADM rule set in 10 (Section 10), you will see that the packet has to pass through your IPFWADM rule sets before being forwarded. Excellent!

- Anyway, download BOTH from the URL in 5 (Section 5)

- ipportfw.c source file - the kernel patch files for 2.0.36

Put this code into the /usr/src directory. I also recommend that you go to Steven's WWW page and copy the "usage" page into a text file on the Linux for future use (there isn't a Man page for IPPORTFW).

- Ok, FTP the latest stable kernel (URL in 5 (Section 5)) to /usr/src/

Update: It should be noted that there is some controversy with putting the Linux kernel sources in /usr/src. Please see `<http://kt.linuxcare.com/kernel-traffic/kt20000814_80.epl##4>` for full details. So, though Linus recommends NOT to /usr/src/linux for new kernels, many programs, patches, etc. assume that the newest kernel sources are in there. Personally, I haven't had any issue with putting the sources in /usr/src/linux but I now use /usr/src/kernel/linux instead.

- Uncompress it (tar -xzvf linux-2.0.36.tar.gz)

- For usability, rename the newly created "linux" direcory to the proper kernel version and then just create a symbolic link to re-create the "linux" directory. e.g.

mv linux linux-2.0.36 ln -s linux-2.0.36 linux

- Copy the IPPORTFW patch into the Linux directory

cp /usr/src/subs-patch-1.37.gz /usr/src/kernel/linux

- Now, you need to patch the kernel for IPPORTFW to become an compilable option:

cd /usr/src/kernel/linux zcat subs-patch-1.3x.gz | patch -p1

- That's it for the kernel for now. Now, compile the IPPORTFW program

cd /usr/src gcc ipportfw.c -o ipportfw

- Finally, install it

mv ipportfw /usr/local/sbin

- If you have additional questions, please see the IP-MASQ-HOWTO found in 5 (Section 5) for FULL details.

12 Initial Linux Kernel compiling

TrinityOS currently reflects the building of both a 2.2.16 and also 2.0.38 kernels. If you didn't already know, Linux kernel numbering follows a rule:

- All EVEN numbered kernels (1.0, 1.2, 2.0, 2.2, 2.4, etc) are all BETA or stable (production) kernels. Beta kernels are usually locked out of having new features added to them so that the developers and concentrate on simply fixing bugs and making the code more stable. Latest numbered kernels are always the best to run.

- All ODD numbered kernels (.9, 1.1, 1.3, 2.1, 2.3, etc) are all ALPHA or test kernels. Alpha kernels are where new Linux features are added, tested, and debugged. After a specific "lockout" period announced by Linus, no more new features can be put into a given Alpha kernel generation. After this, the alpha kernel is simply fixed up for a while more and once the kernel is considered stable, it is moved to the next BETA kernel version and a new ALPHA kernel is started.

Be warned: Alpha kernel revs can be released on occasion that are unstable, cause data corruption, or even not compile at all. Like anything in the Linux world, these issues are fixed at a rapid rate and become more stable every day. As it stands, the latest 2.3.x+ kernels are quite stable and will be rolled into the 2.4.x kernel soon. After this, the 2.5.x Alpha kernel will be started up.

* Anyway, lets get down to compiling up a kernel. All initial steps to getting * the kernel sources and uncompression the kernel is in the previous section [required * since the IPPORTFW patches change the kernel a little]

12.1 Configuring a kernel

There are several ways to configure a kernel:

- Use the command "make config" to configure a kernel the old fashion way
- Use the command "make menuconfig" to configure a kernel via a colorized Ncurses text GUI
- Use the command "make xconfig" to confiure a kernel from an Xwindow GUI

- 2.2.x kernels:

The new 2.2.x kernels are the newer generation in Linus's kernels. They offer enhanced performance, better SMP functionality, etc. At the same time, they had to change some things compared to the 2.0.x kernels and thus broke things. If you are running an older Linux distribution that did NOT come with a 2.2.x kernel, you will have to upgrade at LEAST the following tools:

```
ftp://ftp.rge.com/pub/systems/linux/redhat/updates/5.2/kernel-2.2/i386/

dhcpcd-1.3.16-0.i386.rpm, initscripts-3.78-2.2.i386.rpm, ipchains-1.3.8-0.i386.rpm
modutils-2.1.121-0.i386.rpm, net-tools-1.50-0.i386.rpm, procinfo-15-0.i386.rpm
samba-2.0.0-0.i386.rpm, util-linux-2.9-0.i386.rpm
```

Personally, I highly recommend that you just install an entirely new Linux distribution that natively supports the 2.2.x kernels. This will save you a lot of time and suffering in the long run.

Below configs are for my hardware. Make changes to your config as required

2.2.x kernel setup:

NOTE: This kernel config reflects different hardware than documented in Section 2 of TrinityOS. This kernel is running on a Intel motherboard with:

An Intel Pentium 166Mhz CPU 128MB of RAM (2) 3Com 3c905 PCI Ethernet cards Adaptec 2940U SCSI controller Several IBM and Seagate SCSI HDs Matrox Millentium II PCI video card An additional (2)Serial / (1) Parallel I/O card

12.2 Tricks: Upgrading an existing kernel to a newer one

If you compiled a kernel in the past and got things running fine but now you want to compile up the newest available kernel, there is one cool trick you might want to know about.

Say I compiled up a 2.2.16 kernel on August 12th, 2000.

- What I would do is copy the .config file from the /usr/src/kernel/linux directory (I'm assuming you put the 2.2.16 kernel sources in there) to a safe place such as /usr/src/config/l2216.080100
- Once the the 2.2.17 kernel came out, I would put the new sources into /usr/src/kernel/linux-2.2.17 and create a sym link back pointing to /usr/src/kernel/linux
- From here, I would copy the old 2.2.16 .config file into this new 2.2.17 source directory and rename it back to .config (this is covered in Section 11)
- I would then run the command "make oldconfig" and this will automatically apply all the configuration options from the 2.2.16 kernel to the new 2.2.17 kernel. An additional perk to this script is it will prompt you with any new kernel options

- Once the new 2.2.17 kernel is configured, I would compile it up, and boot it. If it works fine, I would then copy this new .config file to /usr/src/config/l2217.090100.

12.3 A 2.2.16 kernel config

/usr/src/kernel/linux/.config

```
#
# Automatically generated make config: don't edit
#

#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y

#
# Processor type and features
#
# CONFIG_M386 is not set
# CONFIG_M486 is not set
# CONFIG_M586 is not set
CONFIG_M586TSC=y
# CONFIG_M686 is not set
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_TSC=y
CONFIG_1GB=y
# CONFIG_2GB is not set
# CONFIG_MATH_EMULATION is not set
# CONFIG_MTRR is not set
# CONFIG_SMP is not set

#
# Loadable module support
#
CONFIG_MODULES=y
# CONFIG_MODVERSIONS is not set
CONFIG_KMOD=y

#
# General setup
#
CONFIG_NET=y
CONFIG_PCI=y
# CONFIG_PCI_GOBIOS is not set
# CONFIG_PCI_GODIRECT is not set
CONFIG_PCI_GOANY=y
```

```
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_PCI_QUIRKS=y
# CONFIG_PCI_OPTIMIZE is not set
CONFIG_PCI_OLD_PROC=y
# CONFIG_MCA is not set
# CONFIG_VISWS is not set
CONFIG_SYSVIPC=y
# CONFIG_BSD_PROCESS_ACCT is not set
CONFIG_SYSCTL=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_ELF=y
CONFIG_BINFMT_MISC=y
# CONFIG_BINFMT_JAVA is not set
CONFIG_PARPORT=y
CONFIG_PARPORT_PC=y
# CONFIG_PARPORT_OTHER is not set
CONFIG_APM=y
# CONFIG_APM_IGNORE_USER_SUSPEND is not set
# CONFIG_APM_DO_ENABLE is not set
# CONFIG_APM_CPU_IDLE is not set
CONFIG_APM_DISPLAY_BLANK=y
# CONFIG_APM_IGNORE_SUSPEND_BOUNCE is not set
# CONFIG_APM_RTC_IS_GMT is not set
# CONFIG_APM_ALLOW_INTS is not set
# CONFIG_APM_REAL_MODE_POWER_OFF is not set

#
# Plug and Play support
#
CONFIG_PNP=y
# CONFIG_PNP_PARPORT is not set

#
# Block devices
#
CONFIG_BLK_DEV_FD=y
CONFIG_BLK_DEV_IDE=y

#
# Please see Documentation/ide.txt for help/info on IDE drives
#
# CONFIG_BLK_DEV_HD_IDE is not set
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_BLK_DEV_IDECD=y
# CONFIG_BLK_DEV_IDETAPE is not set
# CONFIG_BLK_DEV_IDEFLOPPY is not set
# CONFIG_BLK_DEV_IDESCSI is not set
# CONFIG_BLK_DEV_CMD640 is not set
# CONFIG_BLK_DEV_RZ1000 is not set
```

```
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_BLK_DEV_IDEDMA=y
# CONFIG_BLK_DEV_OFFBOARD is not set
CONFIG_IDEDMA_AUTO=y
# CONFIG_BLK_DEV_OPTI621 is not set
# CONFIG_BLK_DEV_TRM290 is not set
# CONFIG_BLK_DEV_NS87415 is not set
# CONFIG_BLK_DEV_VIA82C586 is not set
# CONFIG_BLK_DEV_CMD646 is not set
# CONFIG_BLK_DEV_CS5530 is not set
# CONFIG_IDE_CHIPSETS is not set

#
# Additional Block Devices
#
CONFIG_BLK_DEV_LOOP=m
# CONFIG_BLK_DEV_NBD is not set
CONFIG_BLK_DEV_MD=y
# CONFIG_MD_LINEAR is not set
CONFIG_MD_STRIPED=y
CONFIG_MD_MIRRORING=y
CONFIG_MD_RAID5=y
CONFIG_MD_BOOT=y
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_SIZE=4096
CONFIG_BLK_DEV_INITRD=y
# CONFIG_BLK_DEV_XD is not set
# CONFIG_BLK_DEV_DAC960 is not set
CONFIG_PARIDE_PARPORT=y
# CONFIG_PARIDE is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_BLK_DEV_HD is not set

#
# Networking options
#
CONFIG_PACKET=y
CONFIG_NETLINK=y
CONFIG_RTNETLINK=y
# CONFIG_NETLINK_DEV is not set
CONFIG_FIREWALL=y
CONFIG_FILTER=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_ADVANCED_ROUTER=y
CONFIG_RTNETLINK=y
CONFIG_NETLINK=y
# CONFIG_IP_MULTIPLE_TABLES is not set
# CONFIG_IP_ROUTE_MULTIPATH is not set
```

```
# CONFIG_IP_ROUTE_TOS is not set
CONFIG_IP_ROUTE_VERBOSE=y
# CONFIG_IP_ROUTE_LARGE_TABLES is not set
# CONFIG_IP_PNP is not set
CONFIG_IP_FIREWALL=y
# CONFIG_IP_FIREWALL_NETLINK is not set
# CONFIG_IP_TRANSPARENT_PROXY is not set
CONFIG_IP_MASQUERADE=y

#
# Protocol-specific masquerading support will be built as modules.
#
CONFIG_IP_MASQUERADE_ICMP=y

#
# Protocol-specific masquerading support will be built as modules.
#
CONFIG_IP_MASQUERADE_MOD=y
# CONFIG_IP_MASQUERADE_IPAUTOFW is not set
CONFIG_IP_MASQUERADE_IPPORTFW=y
# CONFIG_IP_MASQUERADE_MFW is not set
CONFIG_IP_ROUTER=y
# CONFIG_NET_IPIP is not set
# CONFIG_NET_IPGRE is not set
# CONFIG_IP_MROUTE is not set
CONFIG_IP_ALIAS=y
# CONFIG_ARPD is not set
CONFIG_SYN_COOKIES=y

#
# (it is safe to leave these untouched)
#
# CONFIG_INET_RARP is not set
CONFIG_SKB_LARGE=y
# CONFIG_IPV6 is not set

#
#
#
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_X25 is not set
# CONFIG_LAPB is not set
# CONFIG_BRIDGE is not set
# CONFIG_LLC is not set
# CONFIG_ECONET is not set
# CONFIG_WAN_ROUTER is not set
# CONFIG_NET_FASTROUTE is not set
# CONFIG_NET_HW_FLOWCONTROL is not set
# CONFIG_CPU_IS_SLOW is not set
```

```
#
# QoS and/or fair queueing
#
# CONFIG_NET_SCHED is not set

#
# Telephony Support
#
# CONFIG_PHONE is not set
# CONFIG_PHONE_IXJ is not set

#
# SCSI support
#
CONFIG_SCSI=y

#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=y
CONFIG_CHR_DEV_ST=y
CONFIG_BLK_DEV_SR=y
# CONFIG_BLK_DEV_SR_VENDOR is not set
# CONFIG_CHR_DEV_SG is not set

#
# Some SCSI devices (e.g. CD jukebox) support multiple LUNs
#
# CONFIG_SCSI_MULTI_LUN is not set
CONFIG_SCSI_CONSTANTS=y
CONFIG_SCSI_LOGGING=y

#
# SCSI low-level drivers
#
# CONFIG_BLK_DEV_3W_XXXX_RAID is not set
# CONFIG_SCSI_7000FASST is not set
# CONFIG_SCSI_ACARD is not set
# CONFIG_SCSI_AHA152X is not set
# CONFIG_SCSI_AHA1542 is not set
# CONFIG_SCSI_AHA1740 is not set
CONFIG_SCSI_AIC7XXX=y
CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT=y
CONFIG_AIC7XXX_CMDS_PER_DEVICE=8
CONFIG_AIC7XXX_PROC_STATS=y
CONFIG_AIC7XXX_RESET_DELAY=5
# CONFIG_SCSI_IPS is not set
# CONFIG_SCSI_ADVANSYS is not set
# CONFIG_SCSI_IN2000 is not set
```

```
# CONFIG_SCSI_AM53C974 is not set
# CONFIG_SCSI_MEGARAID is not set
# CONFIG_SCSI_BUSLOGIC is not set
# CONFIG_SCSI_DTC3280 is not set
# CONFIG_SCSI_EATA is not set
# CONFIG_SCSI_EATA_DMA is not set
# CONFIG_SCSI_EATA_PIO is not set
# CONFIG_SCSI_FUTURE_DOMAIN is not set
# CONFIG_SCSI_GDTH is not set
# CONFIG_SCSI_GENERIC_NCR5380 is not set
# CONFIG_SCSI_INITIO is not set
# CONFIG_SCSI_INIA100 is not set
# CONFIG_SCSI_PPA is not set
# CONFIG_SCSI_IMM is not set
# CONFIG_SCSI_NCR53C406A is not set
# CONFIG_SCSI_SYM53C416 is not set
# CONFIG_SCSI_SIM710 is not set
# CONFIG_SCSI_NCR53C7xx is not set
# CONFIG_SCSI_NCR53C8XX is not set
# CONFIG_SCSI_SYM53C8XX is not set
# CONFIG_SCSI_PAS16 is not set
# CONFIG_SCSI_PCI2000 is not set
# CONFIG_SCSI_PCI2220I is not set
# CONFIG_SCSI_PSI240I is not set
# CONFIG_SCSI_QLOGIC_FAS is not set
# CONFIG_SCSI_QLOGIC_ISP is not set
# CONFIG_SCSI_QLOGIC_FC is not set
# CONFIG_SCSI_SEAGATE is not set
# CONFIG_SCSI_DC390T is not set
# CONFIG_SCSI_T128 is not set
# CONFIG_SCSI_U14_34F is not set
# CONFIG_SCSI_ULTRASTOR is not set
# CONFIG_SCSI_DEBUG is not set

#
# I2O device support
#
# CONFIG_I2O is not set
# CONFIG_I2O_PCI is not set
# CONFIG_I2O_BLOCK is not set
# CONFIG_I2O_SCSI is not set

#
# Network device support
#
CONFIG_NETDEVICES=y

#
# ARCnet devices
#
```

```
# CONFIG_ARCNET is not set
CONFIG_DUMMY=m
# CONFIG_BONDING is not set
# CONFIG_EQUALIZER is not set
# CONFIG_ETHERTAP is not set
# CONFIG_NET_SB1000 is not set

#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
CONFIG_NET_VENDOR_3COM=y
# CONFIG_EL1 is not set
# CONFIG_EL2 is not set
# CONFIG_ELPLUS is not set
# CONFIG_EL16 is not set
# CONFIG_EL3 is not set
# CONFIG_3C515 is not set
CONFIG_VORTEX=y
# CONFIG_LANCE is not set
# CONFIG_NET_VENDOR_SMC is not set
# CONFIG_NET_VENDOR_RACAL is not set
# CONFIG_RTL8139 is not set
# CONFIG_NET_ISA is not set
# CONFIG_NET_EISA is not set
# CONFIG_NET_POCKET is not set

#
# Ethernet (1000 Mbit)
#
# CONFIG_ACENIC is not set
# CONFIG_HAMACHI is not set
# CONFIG_YELLOWFIN is not set
# CONFIG_SK98LIN is not set
# CONFIG_FDDI is not set
# CONFIG_HIPPI is not set
# CONFIG_PLIP is not set
CONFIG_PPP=y

#
# CCP compressors for PPP are only built as modules.
#
# CONFIG_SLIP is not set
# CONFIG_NET_RADIO is not set

#
# Token ring devices
#
# CONFIG_TR is not set
# CONFIG_NET_FC is not set
```

```
# CONFIG_RCPCI is not set
# CONFIG_SHAPER is not set

#
# Wan interfaces
#
# CONFIG_HOSTESS_SV11 is not set
# CONFIG_COSA is not set
# CONFIG_SEALEVEL_4021 is not set
# CONFIG_SYNCLINK_SYNCPPP is not set
# CONFIG_LANMEDIA is not set
# CONFIG_COMX is not set
# CONFIG_HDLC is not set
# CONFIG_DLCI is not set
# CONFIG_SBNI is not set

#
# Amateur Radio support
#
# CONFIG_HAMRADIO is not set

#
# IrDA (infrared) support
#
# CONFIG_IRDA is not set

#
# ISDN subsystem
#
# CONFIG_ISDN is not set

#
# Old CD-ROM drivers (not SCSI, not IDE)
#
# CONFIG_CD_NO_IDESCSI is not set

#
# Character devices
#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_SERIAL=y
# CONFIG_SERIAL_CONSOLE is not set
# CONFIG_SERIAL_EXTENDED is not set
# CONFIG_SERIAL_NONSTANDARD is not set
CONFIG_UNIX98_PTYS=y
CONFIG_UNIX98_PTY_COUNT=256
CONFIG_PRINTER=m
# CONFIG_PRINTER_READBACK is not set
CONFIG_MOUSE=y
```

```
#
# Mice
#
# CONFIG_ATIXL_BUSMOUSE is not set
# CONFIG_BUSMOUSE is not set
# CONFIG_MS_BUSMOUSE is not set
CONFIG_PSMOUSE=y
# CONFIG_82C710_MOUSE is not set
# CONFIG_PC110_PAD is not set

#
# Joysticks
#
# CONFIG_JOYSTICK is not set
# CONFIG_QIC02_TAPE is not set
# CONFIG_WATCHDOG is not set
# CONFIG_NVRAM is not set
CONFIG_RTC=y

#
# Video For Linux
#
# CONFIG_VIDEO_DEV is not set
# CONFIG_DTLK is not set

#
# Ftape, the floppy tape device driver
#
# CONFIG_FTAPE is not set

#
# Filesystems
#
# CONFIG_QUOTA is not set
CONFIG_AUTOFS_FS=y
# CONFIG_ADFS_FS is not set
# CONFIG_AFFS_FS is not set
# CONFIG_HFS_FS is not set
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
# CONFIG_UMSDOS_FS is not set
CONFIG_VFAT_FS=y
CONFIG_ISO9660_FS=y
CONFIG_JOLIET=y
# CONFIG_MINIX_FS is not set
# CONFIG_NTFS_FS is not set
# CONFIG_HPFS_FS is not set
CONFIG_PROC_FS=y
CONFIG_DEVPTS_FS=y
```

```
# CONFIG_QNX4FS_FS is not set
# CONFIG_ROMFS_FS is not set
CONFIG_EXT2_FS=y
# CONFIG_SYSV_FS is not set
# CONFIG_UFS_FS is not set
# CONFIG_EFS_FS is not set

#
# Network File Systems
#
# CONFIG_CODA_FS is not set
CONFIG_NFS_FS=y
CONFIG_NFSD=m
# CONFIG_NFSD_SUN is not set
CONFIG_SUNRPC=y
CONFIG_LOCKD=y
CONFIG_SMB_FS=y
# CONFIG_NCP_FS is not set

#
# Partition Types
#
# CONFIG_BSD_DISKLABEL is not set
# CONFIG_MAC_PARTITION is not set
# CONFIG_SMD_DISKLABEL is not set
# CONFIG_SOLARIS_X86_PARTITION is not set
# CONFIG_UNIXWARE_DISKLABEL is not set
CONFIG_NLS=y

#
# Native Language Support
#
CONFIG_NLS_DEFAULT="cp437"
CONFIG_NLS_CODEPAGE_437=m
# CONFIG_NLS_CODEPAGE_737 is not set
# CONFIG_NLS_CODEPAGE_775 is not set
# CONFIG_NLS_CODEPAGE_850 is not set
# CONFIG_NLS_CODEPAGE_852 is not set
# CONFIG_NLS_CODEPAGE_855 is not set
# CONFIG_NLS_CODEPAGE_857 is not set
# CONFIG_NLS_CODEPAGE_860 is not set
# CONFIG_NLS_CODEPAGE_861 is not set
# CONFIG_NLS_CODEPAGE_862 is not set
# CONFIG_NLS_CODEPAGE_863 is not set
# CONFIG_NLS_CODEPAGE_864 is not set
# CONFIG_NLS_CODEPAGE_865 is not set
# CONFIG_NLS_CODEPAGE_866 is not set
# CONFIG_NLS_CODEPAGE_869 is not set
# CONFIG_NLS_CODEPAGE_874 is not set
# CONFIG_NLS_CODEPAGE_932 is not set
```

```
# CONFIG_NLS_CODEPAGE_936 is not set
# CONFIG_NLS_CODEPAGE_949 is not set
# CONFIG_NLS_CODEPAGE_950 is not set
CONFIG_NLS_ISO8859_1=m
# CONFIG_NLS_ISO8859_2 is not set
# CONFIG_NLS_ISO8859_3 is not set
# CONFIG_NLS_ISO8859_4 is not set
# CONFIG_NLS_ISO8859_5 is not set
# CONFIG_NLS_ISO8859_6 is not set
# CONFIG_NLS_ISO8859_7 is not set
# CONFIG_NLS_ISO8859_8 is not set
# CONFIG_NLS_ISO8859_9 is not set
# CONFIG_NLS_ISO8859_14 is not set
# CONFIG_NLS_ISO8859_15 is not set
# CONFIG_NLS_KOI8_R is not set

#
# Console drivers
#
CONFIG_VGA_CONSOLE=y
# CONFIG_VIDEO_SELECT is not set
# CONFIG_MDA_CONSOLE is not set
# CONFIG_FB is not set

#
# Sound
#
CONFIG_SOUND=y
# CONFIG_SOUND_CMPCI is not set
# CONFIG_SOUND_ES1370 is not set
# CONFIG_SOUND_ES1371 is not set
# CONFIG_SOUND_MAESTRO is not set
# CONFIG_SOUND_ESSSOLO1 is not set
# CONFIG_SOUND_ICH is not set
# CONFIG_SOUND_SONICVIBES is not set
# CONFIG_SOUND_TRIDENT is not set
# CONFIG_SOUND_MSNDCLAS is not set
# CONFIG_SOUND_MSNDPIN is not set
CONFIG_SOUND_OSS=y
# CONFIG_SOUND_DMAP is not set
# CONFIG_SOUND_PAS is not set
CONFIG_SOUND_SB=y
CONFIG_SB_BASE=220
CONFIG_SB_IRQ=5
CONFIG_SB_DMA=1
CONFIG_SB_DMA2=5
CONFIG_SB_MPU_BASE=330

#
# MPU401 IRQ is only required with Jazz16, SM Wave and ESS1688.
```

```
#

#
# Enter -1 to the following question if you have something else such as SB16/32.
#
CONFIG_SB_MPU_IRQ=-1
# CONFIG_SOUND_GUS is not set
# CONFIG_SOUND_MPU401 is not set
# CONFIG_SOUND_PSS is not set
# CONFIG_SOUND_MSS is not set
# CONFIG_SOUND_SSCAPE is not set
# CONFIG_SOUND_TRIX is not set
# CONFIG_SOUND_VIA82CXXX is not set
# CONFIG_SOUND_MAD16 is not set
# CONFIG_SOUND_WAVEFRONT is not set
# CONFIG_SOUND_CS4232 is not set
# CONFIG_SOUND_OPL3SA2 is not set
# CONFIG_SOUND_MAUI is not set
# CONFIG_SOUND_SGALAXY is not set
# CONFIG_SOUND_AD1816 is not set
# CONFIG_SOUND_OPL3SA1 is not set
# CONFIG_SOUND_SOFTOSS is not set
# CONFIG_SOUND_YM3812 is not set
# CONFIG_SOUND_VMIDI is not set
# CONFIG_SOUND_UART6850 is not set
# CONFIG_SOUND_NM256 is not set
# CONFIG_SOUND_YMPCI is not set

#
# Additional low level sound drivers
#
# CONFIG_LOWLEVEL_SOUND is not set

#
# Kernel hacking
#
# CONFIG_MAGIC_SYSRQ is not set
```

12.4 A 2.0.38 kernel config /w IPPORTFW and LooseUDP patches

/usr/src/kernel/linux/.config

```
#
# Automatically generated by make menuconfig: don't edit
#

#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y
```

```
#
# Loadable module support
#
CONFIG_MODULES=y
# CONFIG_MODVERSIONS is not set
# CONFIG_KERNELD is not set

#
# General setup
#
# CONFIG_MATH_EMULATION is not set
CONFIG_MEM_STD=y
# CONFIG_MEM_ENT is not set
# CONFIG_MEM_SPECIAL is not set
CONFIG_MAX_MEMSIZE=1024
CONFIG_NET=y
# CONFIG_MAX_16M is not set
# CONFIG_PCI is not set
CONFIG_SYSVIPC=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_ELF=y
# CONFIG_BINFMT_JAVA is not set
CONFIG_KERNEL_ELF=y
# CONFIG_M386 is not set
CONFIG_M486=y
# CONFIG_M586 is not set
# CONFIG_M686 is not set
# CONFIG_APM is not set

#
# Floppy, IDE, and other block devices
#
CONFIG_BLK_DEV_FD=y
CONFIG_BLK_DEV_IDE=y
# CONFIG_BLK_DEV_HD_IDE is not set
CONFIG_BLK_DEV_IDECD=y
# CONFIG_BLK_DEV_IDETAPE is not set
# CONFIG_BLK_DEV_IDEFLOPPY is not set
# CONFIG_BLK_DEV_IDESCSI is not set
# CONFIG_BLK_DEV_IDE_PCMCIA is not set
# CONFIG_BLK_DEV_CMD640 is not set
# CONFIG_IDE_CHIPSETS is not set
CONFIG_BLK_DEV_LOOP=m
CONFIG_BLK_DEV_MD=y
CONFIG_MD_LINEAR=y
CONFIG_MD_STRIPED=y
CONFIG_MD_MIRRORING=y
CONFIG_MD_RAID5=y
CONFIG_BLK_DEV_RAM=y
```

```
CONFIG_BLK_DEV_INITRD=y
# CONFIG_BLK_DEV_XD is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_PARIDE is not set
# CONFIG_BLK_DEV_HD is not set

#
# Networking options
#
CONFIG_FIREWALL=y
CONFIG_NET_ALIAS=y
CONFIG_INET=y
CONFIG_IP_FORWARD=y
CONFIG_IP_MULTICAST=y
CONFIG_SYN_COOKIES=y
CONFIG_IP_FIREWALL=y
CONFIG_IP_FIREWALL_VERBOSE=y
CONFIG_IP_MASQUERADE=y
# CONFIG_IP_MASQUERADE_IPAUTOFW is not set
CONFIG_IP_MASQUERADE_IPPORTFW=y
# CONFIG_IP_MASQUERADE_PPTP is not set
# CONFIG_IP_MASQUERADE_IPSEC is not set
CONFIG_IP_MASQUERADE_ICMP=y
# CONFIG_IP_TRANSPARENT_PROXY is not set
CONFIG_IP_MASQ_LOOSE_UDP=y
CONFIG_IP_ALWAYS_DEFRAG=y
# CONFIG_IP_ACCT is not set
CONFIG_IP_ROUTER=y
# CONFIG_NET_IPIP is not set
# CONFIG_IP_MROUTE is not set
CONFIG_IP_ALIAS=y
# CONFIG_INET_PCTCP is not set
# CONFIG_INET_RARP is not set
# CONFIG_NO_PATH_MTU_DISCOVERY is not set
CONFIG_IP_NOSR=y
CONFIG_SKB_LARGE=y
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_AX25 is not set
# CONFIG_BRIDGE is not set
# CONFIG_NETLINK is not set

#
# SCSI support
#
CONFIG_SCSI=y
CONFIG_BLK_DEV_SD=y
CONFIG_CHR_DEV_ST=y
CONFIG_BLK_DEV_SR=y
# CONFIG_CHR_DEV_SG is not set
```

```
# CONFIG_SCSI_MULTI_LUN is not set
CONFIG_SCSI_CONSTANTS=y

#
# SCSI low-level drivers
#
# CONFIG_SCSI_7000FASST is not set
# CONFIG_SCSI_ACARD is not set
# CONFIG_SCSI_AHA152X is not set
# CONFIG_SCSI_AHA1542 is not set
# CONFIG_SCSI_AHA1740 is not set
CONFIG_SCSI_AIC7XXX=y
CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT=y
CONFIG_AIC7XXX_CMDS_PER_DEVICE=8
CONFIG_AIC7XXX_PROC_STATS=y
CONFIG_AIC7XXX_RESET_DELAY=5
# CONFIG_SCSI_ADVANSYS is not set
# CONFIG_SCSI_IN2000 is not set
# CONFIG_SCSI_AM53C974 is not set
# CONFIG_SCSI_MEGARAID is not set
# CONFIG_SCSI_BUSLOGIC is not set
# CONFIG_SCSI_DTC3280 is not set
# CONFIG_SCSI_EATA_DMA is not set
# CONFIG_SCSI_EATA_PIO is not set
# CONFIG_SCSI_EATA is not set
# CONFIG_SCSI_FUTURE_DOMAIN is not set
# CONFIG_SCSI_GENERIC_NCR5380 is not set
# CONFIG_SCSI_INITIO is not set
# CONFIG_SCSI_INIA100 is not set
# CONFIG_SCSI_NCR53C406A is not set
# CONFIG_SCSI_SYM53C416 is not set
# CONFIG_SCSI_PPA is not set
# CONFIG_SCSI_PAS16 is not set
# CONFIG_SCSI_PCI2000 is not set
# CONFIG_SCSI_PCI2220I is not set
# CONFIG_SCSI_PSI240I is not set
# CONFIG_SCSI_QLOGIC_FAS is not set
# CONFIG_SCSI_SEAGATE is not set
# CONFIG_SCSI_T128 is not set
# CONFIG_SCSI_TC2550 is not set
# CONFIG_SCSI_U14_34F is not set
# CONFIG_SCSI_ULTRASTOR is not set
# CONFIG_SCSI_GDTH is not set

#
# Network device support
#
CONFIG_NETDEVICES=y
CONFIG_DUMMY=m
# CONFIG_EQUALIZER is not set
```

```
# CONFIG_DLCI is not set
# CONFIG_PLIP is not set
CONFIG_PPP=y
# CONFIG_SLIP is not set
# CONFIG_NET_RADIO is not set
CONFIG_NET_ETHERNET=y
CONFIG_NET_VENDOR_3COM=y
# CONFIG_EL1 is not set
# CONFIG_EL2 is not set
# CONFIG_ELPLUS is not set
# CONFIG_EL16 is not set
CONFIG_EL3=y
# CONFIG_3C515 is not set
# CONFIG_VORTEX is not set
# CONFIG_NET_VENDOR_SMC is not set
# CONFIG_NET_PCI is not set
# CONFIG_NET_ISA is not set
# CONFIG_NET_EISA is not set
# CONFIG_NET_POCKET is not set
# CONFIG_TR is not set
# CONFIG_FDDI is not set
# CONFIG_ARCNET is not set
# CONFIG_SHAPER is not set
# CONFIG_RCPCI is not set

#
# ISDN subsystem
#
# CONFIG_ISDN is not set

#
# CD-ROM drivers (not for SCSI or IDE/ATAPI drives)
#
# CONFIG_CD_NO_IDESCSI is not set

#
# Filesystems
#
# CONFIG_QUOTA is not set
CONFIG_MINIX_FS=y
# CONFIG_EXT_FS is not set
CONFIG_EXT2_FS=y
# CONFIG_XIA_FS is not set
CONFIG_NLS=y
CONFIG_ISO9660_FS=y
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
# CONFIG_UMSDOS_FS is not set
CONFIG_VFAT_FS=y
```

```
#
# Select available code pages
#
# CONFIG_NLS_CODEPAGE_437 is not set
# CONFIG_NLS_CODEPAGE_737 is not set
# CONFIG_NLS_CODEPAGE_775 is not set
# CONFIG_NLS_CODEPAGE_850 is not set
# CONFIG_NLS_CODEPAGE_852 is not set
# CONFIG_NLS_CODEPAGE_855 is not set
# CONFIG_NLS_CODEPAGE_857 is not set
# CONFIG_NLS_CODEPAGE_860 is not set
# CONFIG_NLS_CODEPAGE_861 is not set
# CONFIG_NLS_CODEPAGE_862 is not set
# CONFIG_NLS_CODEPAGE_863 is not set
# CONFIG_NLS_CODEPAGE_864 is not set
# CONFIG_NLS_CODEPAGE_865 is not set
# CONFIG_NLS_CODEPAGE_866 is not set
# CONFIG_NLS_CODEPAGE_869 is not set
# CONFIG_NLS_CODEPAGE_874 is not set
# CONFIG_NLS_ISO8859_1 is not set
# CONFIG_NLS_ISO8859_2 is not set
# CONFIG_NLS_ISO8859_3 is not set
# CONFIG_NLS_ISO8859_4 is not set
# CONFIG_NLS_ISO8859_5 is not set
# CONFIG_NLS_ISO8859_6 is not set
# CONFIG_NLS_ISO8859_7 is not set
# CONFIG_NLS_ISO8859_8 is not set
# CONFIG_NLS_ISO8859_9 is not set
# CONFIG_NLS_ISO8859_15 is not set
# CONFIG_NLS_KOI8_R is not set
CONFIG_PROC_FS=y
CONFIG_NFS_FS=y
# CONFIG_ROOT_NFS is not set
CONFIG_SMB_FS=y
CONFIG_SMB_WIN95=y
# CONFIG_HPFS_FS is not set
# CONFIG_SYSV_FS is not set
# CONFIG_AUTOFS_FS is not set
# CONFIG_AFFS_FS is not set
# CONFIG_UFS_FS is not set

#
# Character devices
#
CONFIG_SERIAL=y
# CONFIG_SERIAL_PCI is not set
# CONFIG_DIGI is not set
# CONFIG_CYCLADES is not set
# CONFIG_ISI is not set
# CONFIG_STALDRV is not set
```

```
# CONFIG_RISCOM8 is not set
CONFIG_PRINTER=y
# CONFIG_SPECIALIX is not set
# CONFIG_MOUSE is not set
# CONFIG_UMISC is not set
# CONFIG_QIC02_TAPE is not set
# CONFIG_FTAPE is not set
# CONFIG_WATCHDOG is not set
CONFIG_RTC=y

#
# Sound
#
CONFIG_SOUND=y
# CONFIG_PAS is not set
CONFIG_SB=y
# CONFIG_ADLIB is not set
# CONFIG_GUS is not set
# CONFIG_MPU401 is not set
# CONFIG_UART6850 is not set
# CONFIG_PSS is not set
# CONFIG_GUS16 is not set
# CONFIG_GUSMAX is not set
# CONFIG_MSS is not set
# CONFIG_SSCAPE is not set
# CONFIG_TRIX is not set
# CONFIG_MAD16 is not set
# CONFIG_CS4232 is not set
# CONFIG_MAUI is not set
CONFIG_AUDIO=y
# CONFIG_MIDI is not set
CONFIG_YM3812=y
SBC_BASE=220
SBC_IRQ=10
SBC_DMA=1
SB_DMA2=5
SB_MPU_BASE=0
SB_MPU_IRQ=-1
DSP_BUFFSIZE=65536
# CONFIG_LOWLEVEL_SOUND is not set

#
# Kernel hacking
#
# CONFIG_PROFILE is not set
```

- [OPTIONAL – You only need to do this if you have an ancient SoundBlaster-type CDROM drive]

- edit /usr/src/kernel/linux/include/linux/sbpcd.h (as of kernel 2.0.38)

- Roughly at line 77, verify the top most SB address and CDROM port is correct.

- Roughly at line 107, change the "#define DISTRIBUTION" variable to "0" to reflect that you have configured the sound drivers

- Roughly at line 121 and 128, change ALL eject line variable to "0" so the drives won't eject their CDs

Now we need to shift gears and jump to the PPP code installation to verify if there is any newer code in the PPP distribution than the kernel distribution.

- Kernel 2.0.35 didn't come with the new v1.16 3Com driver. Bummer. It was pulled because of problems but I haven't had any and there are a LOT of fixes in it. So, do the following:

- mv /usr/src/kernel/linux/drivers/net/3c509.c /usr/src/kernel/linux/drivers/net/3c509.c.orig

- Download the new driver from:

`<ftp://cesdis.gsfc.nasa.gov/pub/linux/drivers/3c509.c>`

If, for some reason, the drive is not available, email me and I'll mail it to you.

****

13 Compile PPPd

- Download the newest PPP sources from the URL in 5 (Section 5) and put it in "/usr/src"

- "tar -xvzf ppp-2.3.x.tar.gz"

- "cd ppp-2.3.x"

- "configure"

- Now, some patches won't need to be installed based upon the version of PPPD and/or the Linux kernel they are installing.

- "make kernel"

This will update any of the required kernel code to work with this version of PPPd.

- "make"

NOTE: You can use "make USE_MS_DNS=1" to insure your system uses the ISP's offered DNS servers over your statically-configure.

Remember, since TrinityOS will run it's OWN DNS server, it really won't matter.

- "make install"

Ok, now back to the kernel configuring for now.. ===

14 Final Linux Kernel compiling and installation

14.1 Manually compiling the kernel

Time to compile the kernel. You can do it manually via the following commands or use the "built-it" script given below.

```
        "cd /usr/src/kernel/linux"
        "make clean"
        "make dep"
        "make bzImage"
```

---

and allow for the kernel to compile (~3mins on a P-II 233)

- Now, compile and install the necessary system modules:

```
        "cd /usr/src/kernel/linux"
        "make modules"
        "make modules_install"
```

- Once the kernel has compiled, do the following command line (replacing "XYZ" with an identifing name like "2035-masq":

Slackware:

```
                "cp /usr/src/kernel/linux/arch/i386/boot/bzImage /XYZ"
```

Redhat:

```
                "cp /usr/src/kernel/linux/arch/i386/boot/bzImage /boot/XYZ"
```

14.2 Automating kernel compiling via the "build-it" script

If you would like to automate this process in the future, create this script in /usr/src/kernel and run it once you have configured your new kernel.

NOTE: You will want to create the directory /usr/src/kernel/config to store your configured kernel setups. This is a good way to find out what is and isn't enabled in a given kernel.

/usr/src/kernel/build-it

<build-it START>

```
!/bin/sh
#
# Version: 11/10/01
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/~dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates:
#
# 07/09/03 - Added checks to stop the process if the kernel doesn't compile
#          - Added the use of path variables
#          - Added additional echo statements for cleaner output
# 11/10/01 - added the use of mrproper to solve rare kernel module issues
# 11/09/01 - made making "dep" serial as doing via parallel had issues
#          - Holy cow.. forgot to parallelize the making of the kernel
# 10/04/01 - Moved the kernel sources and this script to /usr/src/kernel
# 01/17/00 - Changed the date to use %d over %e and remove
```

```
#            any spacesn the date format.
#          - Changed the layout a little and added some beeps at the end
#

# Multi-process option (enable this even for uni-processor machines..
# seriously)
J=-j4

#Location of the kernel sources
SRC=/usr/src/kernel

# --- Script Body

cd $SRC/linux

#Make sure the $SRC/config directory exists.
cp $SRC/linux/.config $SRC/config/kernel.`date +'%b%d'`

# Deal with rare but troublesome kernel module symbol issues
mv .config ..

echo -e "\n\n**********************************************"
echo -e "**                                           **"
echo -e "**        Pre-Phase 1: make mrproper         **"
echo -e "**                                           **"
echo -e "**********************************************\n\n"
make mrproper



echo -e "\n\n**********************************************"
echo -e "**                                           **"
echo -e "**        Pre-Phase 2: make oldconfig        **"
echo -e "**                                           **"
echo -e "**********************************************\n\n"
mv ../.config .
make oldconfig


echo -e "\n\n**********************************************"
echo -e "**                                           **"
echo -e "**        Pre-Phase 3: make clean            **"
echo -e "**                                           **"
echo -e "**********************************************\n\n"
# Clean up from any previous builds
make $J clean


# Start to time the build time
date > $SRC/kernel-compile-time.`date +'%b%d'`
```

```
#Do not parallelize the DEP phase as it can fail
echo -e "\n\n**********************************************"
echo -e "**                                          **"
echo -e "**       Phase 1/5: make dep                **"
echo -e "**                                          **"
echo -e "**********************************************\n\n"

make dep



# Parallize everything else
echo -e "\n\n**********************************************"
echo -e "**                                          **"
echo -e "**       Phase 2/5: make bzImage            **"
echo -e "**                                          **"
echo -e "**********************************************\n\n"
make $J bzImage

#Did it really compile properly?
if [ ! -f $SRC/linux/arch/i386/boot/bzImage ]; then
   #Send a few beeps
   echo ""
   sleep 1
   echo ""
   sleep 1
   echo ""

   echo -e "\n\n!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
   echo -e "!!                                          !!"
   echo -e "!! ERROR:                                   !!"
   echo -e "!!                                          !!"
   echo -e "!!  Kernel did not properly compile.        !!"
   echo -e "!!  (bzImage file is missing).  ABORTING.   !!"
   echo -e "!!                                          !!"
   echo -e "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n\n"

   #Aborting without cleanup will save a required ojects, etc.
   exit 1
fi

#The kernel binary is present, move on


echo -e "\n\n**********************************************"
echo -e "**                                          **"
echo -e "**       Phase 3/5: make modules            **"
echo -e "**                                          **"
echo -e "**********************************************\n\n"
```

```
make $J modules


echo -e "\n\n**********************************************"
echo -e "**                                          **"
echo -e "**       Phase 4/5: make modules_install    **"
echo -e "**                                          **"
echo -e "**********************************************\n\n"
make $J modules_install


echo -e "\n\n**********************************************"
echo -e "**                                          **"
echo -e "**       Phase 5/5: Move binaries over      **"
echo -e "**                                          **"
echo -e "**********************************************\n\n"

cp $SRC/linux/arch/i386/boot/bzImage /boot/bzImage
cp $SRC/linux/System.map /boot/System.map.new
date >> $SRC/kernel-compile-time.`date +'%b%d'`

echo -e "\n\nCompile Done."
echo -e "\nRename /boot/bzImage to a proper name, edit /etc/lilo.conf,"
echo -e "rename /boot/System.map.new to a proper name, symlink this new"
echo -e "map file to /boot/System.map, and finally and finally re-run "
echo -e "lilo.  Make sure lilo runs cleanly"

#Due to SGML conversions, the ASCII "bell" code might become
# corrupt.  To fix this, edit this file with say Vim, delete the
# "^G" characters and resplace them with the following in INSERT
# mode (the control-q tells Vi to add the following character as
# binary and not ascii:
#
#   Control-Q  Control-G
#
echo ^G
sleep 1
echo ^G
sleep 1
echo ^G
```

<build-it STOP>

Don't forget.. "chmod 700 /usr/src/kernel/build-it"

To run the script, run it as "./built-it"

15 Lilo configuration and installation

Lilo is the typical boot loader for Linux though you don't have to use it. You can also use other loaders like:

- System commander
- Microsoft NT's boot loader
- IBM OS/2's boot loader
- boot into DOS and then use LOADLIN

- Edit the /etc/lilo.conf file to reflect your new kernel.

**NOTE: If you aren't using LILO, you need to configure your boot method (LOADLIN, NT boot loader, OS/2 boot loader, System Commander, etc) to use this new kernel.

**NOTE#2: If you have any DOS LILO entries, I highly recommend to password protect them as shown below.

- Add an entry like below :

```
--
# LILO configuration file
# generated by 'liloconfig'
#
# Start LILO global section
boot = /dev/hda

#My box needs this since I have two 3c509 cards
append="ether=0,0,eth1"

#compact        # faster, but won't work on all systems.
delay = 50
vga = normal    # force sane state
# ramdisk = 0   # paranoia setting
# End LILO global section

# Linux bootable partition config begins
image = /2035-1542-sb16
  root = /dev/hda6
  label = linux
  read-only   # Non-UMSDOS filesystems should be mounted read-only for checking
# Linux bootable partition config ends

other=/dev/hda1
label=dos
password=g3a0uttahere
table=/dev/hda
--
```

Two or more NICs: For a secure system, you should have (2) Ethernet cards installed. One to the cable modem and the other for the internal LAN. If both installed Ethernet cards from different vendors, then skip this next part.

If your two Ethernet cards are identical and you compiled support for them into the kernle, Linux will only autodetect ONE card. To make Linux look for additional Ethernet cards, add the following to the lilo.conf file:

```
append="ether=0,0,eth1"
```

If you are using Redhat's dynamic kernel modules to support your network cards, do the following instead:

```
/etc/conf.modules
--
alias eth1 3c509
--
```

This says eth1 is a 3Com 3c509. If it uses non-standard addresses, IRQs, etc, you can specify their locations:

```
/etc/conf.modules
--
options 3c509 io=0x300,12
--
```

Missing Memory: When you boot your machine and run a "dmesg" or a "free" and you don't see all your installed RAM, do the following. This example is for a system with 40MB of RAM..

```
/etc/lilo.conf
--
append="mem=40M"
--
```

- Run the LILO program by simply entering "lilo" at the command prompt to re-write your boot sector. If everything is ok, you will be given a short list of boot images that LILO will boot from.

Before you reboot your box, I *highly* recommend you create a boot disk that will use the kernel off the diskette BUT mount your Linux partition on the hard drive. A RESCUE diskette will NOT let you fix LILO problems. Sucks but its true!

Additional Security: LILO has a feature to password itself. Without the password given, the machine will boot into its configured kernel image. To enable this, edit in the following:

```
/etc/lilo.conf
--
restricted
password=xxxx
--
```

Change the "xxx" to a password of your choice. The "restricted" word enables the passwording. Since the password is saved in CLEAR-TEXT, make sure no one else can read it by doing the following:

```
chmod 700 /etc/lilo.conf
```

LILO booting problems?

"LI" - Getting this when you are rebooting? This realistically is happening because the hard drive geometry in the CMOS setup is different than reported by the kernel booting up. To fix this, add the following line after the "VGA=normal" line:

```
/etc/lilo.conf
--
linear
--
```

If this doesn't help you, check out the LILO docs. Its kinda long but you can just skip down to roughly 93% of it and see what all the LILO codes mean.

```
/usr/doc/lilo-*/README
```

16 Additional RC script configuration and TCP/IP network optimization

Since my system uses all (4) COMM ports and Linux doesn't like to share interrupts (IRQs), you have to tell Linux how to use your specific hardware setup. In addition to configuring Linux to understand your hardware setup, you need to optimize it for maximum performance (serial ports, etc).

NOTE: Until I added these changes, both GPM (tty mouse program) and Xwindows (Xfree86, MetroX, etc) would not load correctly let alone be useful.

16.1 Serial Port Optimizations:

———————————

NOTE: Starting with later 2.1.x and 2.2.x kernels, you do NOT have to set up the follow parameters to get 115,200 on serial ports. If you call the ports via Minicom, PPP, etc at 115,200, it will just work!!

BUT, by setting these files up, any application that asks for 38,400 will actually get 115,200.

For 2.2.x and 2.0.x kernels

/etc/rc.d/rc.serial file:

```
--
#!/bin/sh

SETSERIAL="/bin/setserial -b"

echo "Configuring COM1 for 115200"
${SETSERIAL} /dev/ttyS0 spd_vhi

#echo "RE-configuring COM3 and COM4 to use proper IRQs"
#${SETSERIAL} /dev/ttyS2 uart 16450 port 0x3E8 irq 3
#${SETSERIAL} /dev/ttyS3 uart 16550A port 0x2E8 irq 5

${SETSERIAL} -bg /dev/ttyS0 /dev/ttyS1 /dev/ttyS2 /dev/ttyS3

echo "rc.serial done."
--<end>--
```

Make it executable

```
chmod 700 /etc/rc.d/rc.serial
```

Redhat:

Do a search for "rc.serial" in the /etc/rc.d/rc.sysinit file. If it isn't there, add it at the bottom.

```
/etc/rc.d/rc.sysinit
--
# Initialize the serial subsystem
/etc/rc.d/rc.serial
--
```

Since I use an older Logitech C7 mouse, Linux doesn't come on-line with it the first time. Edit this to suit your hardware configs.

Fix this by doing:

Redhat: Edit /etc/rc.d/init.d/gpm

replace this:

```
daemon gpm -t $MOUSETYPE
```

with this:

```
daemon gpm -b 9600 -r 50 -t $MOUSETYPE
```

Slackware: Edit /etc/rc.d/rc.local

replace this:

```
gpm -t logi
```

with

```
gpm -b 9600 -r 50 -t $logi
```

16.2 Network Optimization:

16.2.1 Ethernet NIC

Vendor Specific: Most 3Com Ethernet ISA and PCI NICs have a ————- DOS based utility that allows you to enable/disable Plug and Play, manually configure IO ports, IRQs, and specify both the IRQ utilization and priority.

Personally.. I always recommend to DISABLE Plug and Play and manually configure the cards as depicted in 4 (Section 4). Anyway, I also recommend the following:

Serial-attached analog/isdn modem users:

- Set your Ethernet cards to support a modem IRQ utiliztion for 19200 or faster

- Set your NIC optimization for SERVER

Ethernet Router/cable-modem users:

- Set your Ethernet cards to for NO modem

- Set your NIC optimization for SERVER

—- Brief Overview:

- The Modem speed section tells the Ethernet card NOT to hog the IRQ lines too much. Though most PC serial ports have 16550 or better chipsets, if the serial port is ignored for too long, data will be lost.

- The Optimization field tells the NIC how to utilize things like IRQ duration, DMA bus retention, etc. The Server setting will optimize the NIC for fastest performance at the detriment of CPU utilization. This is the BEST setting for Linux boxes that are doing IP Masq, routing, etc.

16.2.2 TCP/IP Stack specific:

Both Slackware and Redhat, out of the box, do NOT optimize the TCP/IP window size. This can make a BIG difference with performance. For more information, check out URLs in 5 (Section 5):

RFC 1106 - High Latency WAN links - Section 4.1

RFC 793 - Transmission Control Protocol

NOTE to DHCP users:

- You will notice that if you run **/sbin/netstat -rn** and look in the "window" column, your DHCPed interfaces will NOT have an optimal TCP window setting (only worry about the valid IP addresses and NOT the network addressed entries). Neither dhcpcd nor pump have an option to set the window size and I'm not sure about dhclient. I'm still looking for an elegant solution to this so if you have som ideas, let me know.

Redhat:

NOTE: Users that have NOT installed the initscripts-3.67-1.i386.rpm patch RPM, the correct line numbers will be 119 and 134. Personally, I recommend that you just install the RPM NOW!

Edit "/etc/sysconfig/network-scripts/ifup" and around lines 134, 136, 141, 149, and 158, find the lines:

```
        line 134 for Redhat 5
                        or
        line 157 for Mandrake 7:

        "route add -net ${NETWORK} netmask ${NETMASK} ${DEVICE}"

                to:

      "route add -net ${NETWORK} netmask ${NETMASK} window 16384 ${DEVICE}"

Next..


        line 136 for Redhat 5
                        or
        line 157 for Mandrake 7:

        "route add -host ${IPADDR} ${DEVICE}"
```

```
                to:

        "route add -host ${IPADDR} window 16384 ${DEVICE}"


Next...

        line 141 for Redhat 5
                or
        line 162 for Mandrake 7:

        "route add default gw ${GATEWAY} metric 1 ${DEVICE}"

                to:

        "route add default gw ${GATEWAY} window 16384 metric 1 ${DEVICE}"

Next..

        line 149 for Redhat 5
                or
        line 170 for Mandrake 7:

        "route add default gw ${GATEWAY} ${DEVICE}"

                to:

        "route add default gw ${GATEWAY} window 16384 ${DEVICE}"

Next...

        line 158 in Redhat 5
                or
        line 173 in Mandrake 7

        "route add default gw $gw ${DEVICE}"

                to:

        "route add default gw $gw window 16384 ${DEVICE}"
```

Slackware:

Edit /etc/rc.d/rc.inet1" and around lines 47 and 49, find the following text (note: your setup might look a little different so make any changes that are needed for your setup)

```
        "/sbin/route add -net ${NETWORK} netmask ${NETMASK} eth0"
                and
        "if [ ! "$GATEWAY" = "" ]; then
           /sbin/route add default gw ${GATEWAY} netmask 0.0.0.0 metric 1
```

```
        fi"
```

---

and replace them with the following:

```
        "/sbin/route add -net ${NETWORK} netmask ${NETMASK} window 16384 eth0"
                and
        "if [ ! "$GATEWAY" = "" ]; then
   /sbin/route add default gw ${GATEWAY} netmask 0.0.0.0 window 16384 metric 1
        fi"
```

After everything is set and you either run these commands manually or reboot, a "netstat -rn" should look something like:

```
--
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
100.200.0.0     0.0.0.0         255.255.255.0   U      1500 16384      0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U      3584 0          0 lo
0.0.0.0         100.200.0.1     0.0.0.0         UG     1500 16384      0 eth0
--
```

Also, in a pinch, if you need an example of how to address a NIC, say eth1 in Redhat-speak, here is how you do it:

```
        /etc/sysconfig/network-scripts/ifcfg-eth1
        --
        DEVICE=eth1
        IPADDR=192.168.0.1
        NETMASK=255.255.255.0
        NETWORK=192.168.0.0
        BROADCAST=192.168.0.255
        ONBOOT=yes
        BOOTPROTO=none
        --
```

17 Patching, Compiling, and installing IPFWADM

NOTE: This is only needed for 2.0.x kernels. 2.2.x kernel users will need to use IPCHAINS which usually is already installed in modern distribution. It can also be found at a URL in 5 (Section 5)

- FTP the ipfwadm source code tgz or RPM file to "/usr/src/"

- Un-compress the IPFWADM tgz file ("tar -xzvf ipfwadm-2.3.0.tgz") or install the RPM file ("rpm -i ipfwadm-2.3.0-1.i386.rpm")

Note: If you already installed IPFWADM and the above RPM installation didn't work, don't worry, the stock IPFWADM that comes with Redhat will work ok.

- FTP the IPFWADM timeout patch to /usr/src/ipfwadm-2.3.0

- Un-compress the IPFWADM patch ("gunzip ipfwadm-2.3.0-generic-timeout.patch.gz")

- Apply the timeout patch "patch -p0 < ipfwadm-2.3.0-generic-timeout.patch"

- Make sure that all "Hunks Succeed"

- Edit the "ipfwadm.c" file

- At line 107, insert this line:

```
                #include <linux/timer.h>
```

- Compile IPFWADM by doing:

```
        "make"
        "make install"
```

18 Mail aliases for system administration

If you rarely login as root on this Linux server but you *DO* login or read email on another account, I recommend to redirect your "root" mail to that email address.

Please see the Sendmail documentation in 25 (Section 25) on the various changes to Sendmail over the various versions but for now, do the following:

```
    Sendmail - 8.9.x :     /etc/aliases
            or
    Sendmail - 8.1x.x :    /etc/mail/aliases
```

To do this, change the line towards the bottom of the file

- **NOTE:** If you want to have a given email go to MULTIPLE email addresses, simply #'ed out the following line and then create the file ˜root/.forward. In this file, put all of the desired to-be-forwarded email addresses in this file (one email address per line).

Edit the /etc/aliases file and insert the following lines after the "root" line towards the bottom if you have YOUR OWN DOMAIN and run the Sendmail daemon:

```
                #If you have your own domain name and run DNS
                hostmaster: root

                #If you run a WWW site
                webmaster: root

                #If you have your own domain and run email servers
                postmaster: root
                abuse: root

                #For example: root: johndoe@acme123.com
                root: your-final-destination-email-address
```

Now you need to compile up this new alias file by running the command "newaliases". If you get a warning about duplicated lines, simply remove the duplicate lines and re-run "newaliases".

NOTE: If you are running a older version of Sendmail.. I could tell you how to fix your aliasing issues BUT, I'm going to make you upgrade your version of Sendmail! There are so many security issues with older versions of Sendmail that it's just not worth it.

NOTE-2: Please note that if this machine will be acting as a SECONDARY mail server for other Internet domains, you need to know about possible conflicts between the `/etc/mail/local-host-names` and `/etc/mail/aliases` files. Please see 25 (Section 25) for all the critical details.

19 Preparing for reboot and clearing the logs

- For trouble shooting, do the following:

Slackware:

```
"mv /var/adm/messages /var/adm/messages.old"
"touch /var/adm/messages"
"mv /var/adm/syslog /var/adm/syslog.old"
"touch /var/adm/syslog"
"mv /var/adm/debug /var/adm/debug.old"
"touch /var/adm/debug"
```

Redhat:

```
"mv /var/log/messages /var/log/messages.old"
"touch /var/log/messages"
"mv /var/log/syslog /var/log/syslog.old"
"touch /var/log/syslog"
"mv /var/log/debug /var/log/debug.old"
"touch /var/log/debug"
```

- Reboot with the new kernel

- Once the computer has rebooted, look at both (substitute [xxx] for either "log" or "adm" for your respective Distro) the /var/[xxx]/messages and /var/xxx]/syslog files to make sure no errors or problems were found. If there were errors.. fix them before you continue.

20 Verifing MASQ module installation

If you setup IP Masq, make sure that the MASQ modules have loaded.

- make sure all of the IP MASQ modules are running by typing in "lsmod"

- You will see the following:

```
roadrunner:/usr/src/ppp-2.2.0g# lsmod
Module:        #pages:  Used by:
ip_masq_raudio     1           0
ip_masq_quake      1           0
ip_masq_irc        1           0
ip_masq_ftp        1           0
bsd_comp           1           0
```

** If you don't see *ALL* of these, check your /etc/rc.d/rc.modules and try loading them manually by doing "./etc/rc.d/rc.modules"

21 Install TCPDUMP

TCPDUMP is loaded by default in most modern Linux distributions. If it isn't installed, you can get it from the URL in 5 (Section 5)

TCPDUMP–

- Download the "libpcap" source and run the following commands:

```
        "md5sum libpcap-x.y.z.tar.gz" (exchange the x.y.z for your
version)
        <bf>verify that this md5 hash is the same as the one posted from the
        libpcap URL in <red id="sect-5" name="Section 5">
        <p>
        run "./configure"
        "make"
        "make install"
        "make install-man"
        "make install-incl"
        "cp libpcap/bpf/net/* /usr/include/net"
```

- Download "tcpdump" and do the following commands:

```
        "md5sum tcpdump-x.y.z.tar.gz" (exchange the x.y.z for your
version)
        <bf>verify that this md5 hash is the same as the one posted from the
        tcpdump URL in <red id="sect-5" name="Section 5">
        <p>
        "configure"
        "make"
        "make install"
        "make install-man"
```

- Now run "tcpdump" and watch it fly. Look at TCPDUMP's man page as you can send captures to a file, filter the traffic to only stuff you care upon based on source IP, destination IP, ports, UDP, TCP, etc.

22 PPPd configuration [For both PRIMARY and BACKUP PPP connections]

22.1 Thoughts on PPP and its Dial-on-Demand feature

This PPP section is intended for the use of a MANUAL PPP connection for both:

- Users to configure PPPd to dial out to the Internet as their PRIMARY link

- Users to configure PPPd to dial out to the Internet as a BACKUP link

Dial-On-Demand style PPP connections are documented in TrinityOS in the 23 (Section 23 - DialD) section. Though recent versions of PPPd versions support Dial-On-Demand functionality, it hasn't been as flexible as Diald but this is no longer the case. The newest versions of PPPd support full filtering of interesting/non-interesting packets to keep the line down or up. Because of this, I would recommend to simply just use PPPd instead of Diald. Though I need to expand this section, here are a few pro/con sections:

Anyway, regardless of your PPP use, you have a PPP enabled kernel running. This is fully described in 12 (Section 12)

——

Notes for people thinkink of using Multi-Link PPP (ML/PPP) for multiple connections to the same remote site:

As of 01/22/00, the ML/PPP code is moving quite well. Some are patches to PPPd while others are not. Most patches are only for 2.2.x kernels and have issues. Here is an email I receive about one user's view:

```
-- From Charles @ chas@pcscs.com

>This link: http://mp.mansol.net.au/
> is not available as of the time of this mailing.
>
> It does, however, have functional mods for kernels 2.2.13 and 2.2.14. I
> have worked with the 2.2.13 kernel and have been pleased with the
> functionality, but I would say that the code is not ready for production
> machines as there are still latency issues as well as overhead issues with
> 3 or more links in a bundle- at least from my observations. With 3 lines,
> the latency was jumping from 150ms to 750ms.  With 2 lines, the latency
> was smoother with ranges of 150ms to 300ms, but rarely perfect.
>
> There are also
> fault tolerance issues with automated link resets and bundling. If one
> maintains the individual links manually, however, this is a functional
> solution, but by no means an installation which you can walk away from for
> long periods of time and guarantee fault tolerance. Novell's NIAS is still
> the best I have seen in these regards as it meets the demands if high load
> in both large and small packet fills.
>
> For Linux, Chris Pascoe's code is by far the most evolved code I have seen.
> He shows great promise of mature code in a relatively short period of time.
> He has also shown integration with the ppp daemon and ppp kernel
> architecture to be an effective way for doing asynchronous analog and
> synchronous adapter-based MLPPP. There are rumors and controversy with
> regards to modifying Linux PPP's architecture altogether to streamline
> features of MLPPP, asynchronous analog and synchronous PPP links for better
> uniformity. In my opinion, however, Chris' technique is going to be more
> compatible for hardware functionality than an architectural PPP rebuild
> that reduces feature modularity in its design.
>
> As far as the final production stuff:
> If you want performance, you are going to need features such as data and/or
```

```
> VJ header compression for PPP packets. I haven't seen Linux code support that
> yet. I also haven't seen Linux code handle link bundling perfectly yet.
> Links seem to add well and some links can even go down, but there are still
> issues with the 1st link going down causing the whole bundle to need to be
> reset via killall pppd. These refinements, I'm sure, will be last on the
> "TO DO" list and will probably be quite some time before they are properly
> implemented, nevertheless, Linux does in fact now support MLPPP.

>>I also haven't seen Linux code handle link bundling perfectly yet.
>>Links seem to add well and some links can even go down, but there are
>>still issues with the 1st link going down causing the whole bundle to need to
>>be reset via killall pppd. These refinements, I'm sure, will be last on
>>the "TO DO" list and will probably be quite some time before they are
>>properly implemented, nevertheless, Linux does in fact now support MLPPP.

-----
```

Anyway, for you Normal PPP users, here is the TrinityOS setup.

/etc/ppp/chat.your-ppp-isp

```
--
ABORT BUSY ABORT 'NO CARRIER' "" ATZ OK ATM0S11=40 OK ATDT5551212 CONNECT ""
--
```

Fix its permissions: chmod 600 /etc/ppp/chat.your-ppp-isp

```
-- /etc/ppp/pap-secrets
*       your-ppp-login  your-ppp-password
--
```

Fix its permissions: chmod 600 /etc/ppp/pap-secrets

/etc/ppp/options

```
--
# MTU settings will greatly effect your performance, please read up
# on calculating MTU settings from my PPP web page.
# <url url="http://www.ecst.csuchico.edu/~dranch/PPP/ppp-performance.html#mtu">
#
# This setup is optimized for file transfers and NOT for interactive
# traffic like telnet, talk, etc
#
#       14.4k modem users:              296
#       28.8/33.6k modem users:         470
#   IP Masq users (regardless of speed): 1500


# Masq users: If you get a lot of "MASQ: failed TCP/UDP checksum for
#             xxx.xxx.xxx.xxx" errors, turn off VJ header compression
#             by do the following:
```

```
#
# -vj

#pppd v2.3.x PAP config
require-pap

#Get a dynamic IP address.  If you have a static IP addres, put
# the static IP address in the LEFT hand address space
0.0.0.0:0.0.0.0

asyncmap 0
lock
#Use Hardware flow control
crtscts
#BSDComp is a more modern compression method than "deflate"
bsdcomp 15,15
lcp-restart 1
ipcp-restart 1
defaultroute

#Enable these for debugging
#debug
#kdebug 1

user your-ppp-login
--
```

Fix its permissions: chmod 600 /etc/ppp/options

/usr/local/sbin/startppp

```
--
#!/bin/sh
#
# Version: 07/03/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# NOTE:  This configuration assumes that your modem is on COM2
#

echo Killing any stray PPPD processes
killall pppd
killall chat
echo Beginning PPP negotiation..

#Replace /dev/ttyS1 with your modem's COMM port. Remember, always start
```

```
#counting with "0".  Also, make SURE that the paths for pppd/chat are
#in /usr/sbin.  If not, change this command line to use the correct path
#Old pppd v2.2.x format

#New pppd v2.3.x format
/usr/sbin/pppd /dev/ttyS1 38400 crtscts -d lock defaultroute connect '/usr/sbin/chat -v -t 45 -f /etc/ppp/chat.yo
--
```

Fix its permissions: chmod 700 /usr/local/sbin/startppp

/usr/lib/ppp/stopppp

```
--
#!/bin/sh
#
# Version: 07/03/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# NOTE:  This configuration assumes that your modem is on COM2
#

echo Shutting down PPP
#
#Replace /dev/ttyS1 with your modem's COMM port.. remember, always start
#counting with "0".  Also.. make SURE that the paths for pppd/chat are
#in /usr/sbin.  If not, change this command line to use the correct path

/usr/lib/ppp/pppd /dev/ttyS1 disconnect
echo Killing any stray PPPD processes
killall chat
killall pppd
--
```

Fix its permissions: chmod 700 /usr/local/sbin/stopppp

22.2 Primary PPP users using Strong Firewalls:

If you are using the strong firewall rule sets (IPCHAINS/IPFWADM), you will need to re-run your firewall rule set everytime you get your dynamic IP address. To do this:

- Edit or create the file called /etc/ppp/ip-up and in it put:

```
--
#!/bin/sh
/etc/rc.d/rc.firewall
```

```
                #OPTIONAL:  Its nice to be able to update your system
                #              clock when on-line.  To do this, add these
                #              lines, un # them out, and then follow the
                #              instructions in TrinityOS <ref id="sect-26" name="Section 26">
                #
                #       /usr/local/bin/getdate
                --
```

- now fix the permissions on it:

```
                chmod 700 /etc/ppp/ip-up
```

That's IT!

Backup PPP links: If you are like me, you either have a locked up ADSL or Cablemodem connection to the Internet. Well, from time to time, your connection will go down for various reasons and you'll be SOL for Internet access.

What can you do? Setup a backup PPP link! Currently, the config shown below will need to be invoked MANUALLY. It is my plan that once I received my ISDN line, I will develop an AUTOMATIC dial-backup configuration based upon a series of connectivity criteria that will be put into the Diald section of TrinityOS.

NOTE: This rule set is OLD and isn't nearly are secure as the new IPCHAINS rule set found in 10 (). I hope to either port a version of the strong IPCHAINS rule set here soon or make the master rule set adapt to changing environments.

NOTE: When your primary link goes down, your old /etc/rc.firewall rule set will NOT let you out (changed external IP address). So, you need to enter in the following files to bring-up and bring-down a temporary firewall.

/etc/ppp/ip-up

```
--
#!/bin/sh

echo "Starting /etc/ppp/ip-up"

# ----------------------------------------------------------------------------------
#       NOTE:  This short firewall script is for IPFWADM (2.0.x kernels) to only allow
#                      SSH, DNS, and NTP in or out of the PPP0 connection.  If you need additional
#                      connectivity, go ahead and add them in.
#


#Specification of the LOOPBACK interface
loopback="127.0.0.1"

#Specification of the INTERNAL NIC
intif="eth1"

#The IP address on your INTERNAL nic
intip="192.168.0.1"
```

```
#IP network address of the INTERNAL net
intnet="192.168.0.0"

#IP address of an internal host that should have IPPORTFW forward traffic to
portfwip="192.168.0.20"


#Specification of the EXTERNAL NIC
#
#       PPP Users: If you are using the Dynamic PPP "extif" script from above,
#               make sure to comment the below line out so it doesn't override it.
#
#               If you want to use the PPPd variables, change this to read:
#                       extif="$1"
#
extif="ppp0"

#The IP address you get from the Internet
#
#       PPP users: If you are getting dynamic address, either use the "extip" script
#                       from the header above or if you want to use the PPPd variables,
#                       change this to read:
#                       extip="$3"
#
extip="100.200.0.212"

# The IP broadcast address of the external net
#
#       PPP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "extbroad" to read (this make an assuption but it should
#                       be a safe assumption):
#                       extbroad=`echo $4 | cut -d '.' -f 1-3`.255
#
extbroad="100.200.0.255"

#IP address of the default gateway on the EXTERNAL NIC
#
#       PPP users: If you are getting dynamic address, use the PPPd variables.
#                       Change "dgw" to read:
#                       dgw=$4
#
dgw="100.200.0.1"

#IP Mask for ALL IP addresses
universe="0.0.0.0"

#IP Mask for BROADCAST
broadcast="255.255.255.255"

#Specification of HIGH IP ports
```

```
#   NOTE: Notice that this STARTS at 1024 and NOT at 1023 which it should.
#         for some reason SSH sometimes initiates connections at 1023 which
#         is a TCP violation but shit happens.
#
#   Brief update:  This is due to SSH not being executed with "-P"
#
unprivports="1024:65535"

#Specification of backup DNS server
secondarydns="102.200.0.25"

#Specifically allowed external host - secure1.host.com
securehost="200.211.0.40"


# ----------------------------------------------------------------------------------

echo "Change default route to PPP"
/sbin/route add default gw $dgw

echo "Enabling IP Forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward

echo "Changing IP MASQ Timeouts.."
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
#  60 sec timeout for UDP traffic (MASQ'ed ICQ users must enable a 30sec
#                                              firewall timeout in ICQ itself)
/sbin/ipfwadm -M -s 7200 10 60

#Flush all old rule sets
echo "Flushing old poicies"
/sbin/ipfwadm -I -f
/sbin/ipfwadm -O -f
/sbin/ipfwadm -F -f

#Change default policies
echo "Setting default policies to REJECT"
/sbin/ipfwadm -I -p reject
/sbin/ipfwadm -O -p reject
/sbin/ipfwadm -F -p reject

echo "Allow SSH DNS through the PPP0 interface"
/sbin/ipfwadm -I -i accept -W $extif -P tcp -S $universe/0 -D $extip/32 ssh domain ntp
/sbin/ipfwadm -I -i accept -W $extif -P udp -S $universe/0 -D $extip/32 domain

echo "Allow ICMP through the PPP0 interface"
/sbin/ipfwadm -I -i accept -W $extif -P icmp -S $universe/0 -D $extip/32

echo "Allowing SSH, DOMAIN, and ICMP out"
```

```
/sbin/ipfwadm -O -i accept -W $extif -P tcp -S $extip/32 $unprivports -D $universe/0 ssh domain ntp
/sbin/ipfwadm -O -i accept -W $extif -P udp -S $extip/32 $unprivports -D $universe/0 domain
/sbin/ipfwadm -O -i accept -W $extif -P icmp -S $extip/32 -D $universe/0

echo "Masquerade from local net on local interface to anywhere."
/sbin/ipfwadm -F -a masquerade -W $extif -S $intnet/24 -D $universe/0

echo "Logging all failed connections"
/sbin/ipfwadm -I -a reject -S $universe/0 -D $universe/0 -o
/sbin/ipfwadm -O -a reject -S $universe/0 -D $universe/0 -o
/sbin/ipfwadm -F -a reject -S $universe/0 -D $universe/0 -o

echo "Temporary PPP0 firewall and MASQ Done.
--
```

/etc/ppp/ip-down

```
--
#!/bin/sh

# Re-run the master firewall rule set to reset the firewall back to the primary
# interface.

/etc/rc.d/rc.firewall

# /sbin/route add default gw 24.1.83.1

LOGDEVICE=$6
REALDEVICE=$1

[ -x /etc/ppp/ip-down.local ] && /etc/ppp/ip-down.local $*

/etc/sysconfig/network-scripts/ifdown-post ifcfg-${LOGDEVICE}

exit 0
--
```

22.3 FAQ: PPP issues and troubleshooting

- If you get the following error:

```
    Jun  6 21:12:18 server chat[499]: Can't get terminal parameters: Input/output error
    Jun  6 21:12:18 server pppd[498]: Connect script failed
```

  This probably means that PCMCIA services aren't running. Start them up by running:

  Redhat: /etc/rc.d/init.d/pcmcia start

- This was sent from a user who had PPP0 running but it would fault:

```
--
from: Donald Spoon" <marsala@txdirect.net>

The Microsoft web-site, and Stroud's Consummate Winlist web-site would
literally take MINUTES to load!  I had others that exhibited similar
behavior, mainly in the *.mil domain, but most sites would load
fairly quickly as expected.  I played around with the MTU / MRU settings and
found an "optimum" set-up for me that helped a great deal, but the
"selective" delay in loading certain web sites remained.  One day I noticed
that when I had brought the PPP link up MANUALLY the
affected web-sites loaded normally!!
I did one more review of your notes and applied the
suggestions for re-setting lcp-restart = 1, and ipcp-restart = 1 (from
defaults of 3 in the /etc/ppp/options file!.  This change alone did the
trick for  me!
--
```

23 Diald [For Modem users only]

Diald is a mechanism that will do auto-dialing and auto-PPP negotiations for Linux.

It needs to be mentioned that in the past, the PPPd code could do Dial-on-Demand but it wasn't very flexible. This is no longer the case. PPPd now has the same strengths as Diald in the respect to understanding what traffic should bring the line up, keep the line up, or not be counted to then let the line hang up. Because of this, I recommend to ** NOT USE Diald ** anymore.. use PPPd directly. If you have points to why you disagree, please let me know.

Unfortunately, Dial-on-Demand for PPPd isn't documented in TrinityOS yet so you are on your own for now. If you need help, email me but beyond that, Diald should work fine as well.

NOTE: Diald now has a new maintainer and has been updated to v0.98. The the URLs are in 5 (Section 5)

```
+--------------------------------------------------------------------------+
| Follow this link for more information until I can integrate it into the |
| TrinityOS doc:                                                           |
|                                                                          |
|   http://www.ecst.csuchico.edu/dranch/PPP/ppp-performance.html#linux |
+--------------------------------------------------------------------------+
```

Here are a few quick tips:

Use dcntrl or diald-top to see what networ traffic is bringing up your PPP/SLIP link.

Rough order to get things running:

```
- /etc/rc.d/rc.S
        Enabled rc.serial load up

- /etc/rc.d/rc.serial
        /bin/setserial /dev/ttyS1 spd_vhi

cp diald.conf /etc/diald
```

```
diald.conf:
--
restrict 16:00:00 20:45:00 * * *
down
restrict * * * * *
mode ppp
connect /etc/ppp/diald/earthlink-connect
device /dev/cua1
speed 115200
modem
lock
crtscts
local 192.168.1.7
remote 0.0.0.0
dynamic
defaultroute

accounting-log /var/adm/ppp.log
include /usr/local/lib/diald/standard.filter
--
```

In /etc/rc.d/rc.local, add the following line:

```
--
cat "1" > /proc/sys/net/ipv4/ip_dynaddr
```

24 DNS: Acquiring and configuring CHROOTed and SPLIT master/slave DNS servers

The daemon called "named" is the DNS or "Domain Name Server" service that converts Internet hostnames like "www.yahoo.com" to IP addresses like 204.71.177.71 (one of Yahoo's MANY TCP/IP addresses). Though there are other DNS server alternatives to ISC's BIND, it is the most common and best maintained version available. As you might have already figured out, this is a CRITICAL service for the Internet.

TrinityOS documents how to setup multiple Internet domains for full TCP/IP address subnets using both Bind9 and Bind8. It also also covers advanced redundancy and security topics such as remote secondary (backup) DNS servers and both "*CHROOTed Jails*" and "*Split Zone*" files. For the time being, TrinityOS does NOT currently cover Dynamic DNS or DNSSEC. These topics will be covered in future revisions.

What are some of these advanced topics?

- The *CHROOTed* feature means that the named daemon which runs usually as the "root" user will run in its own isolated area. This behavior is very similar to the access that an anonymous FTP user gets when they log into a server and can only see a subset of the remote file system. The main reason to implement this feature is that if some new named security exploit comes out and a hostile user (cracker) finds your machine, they will be extreamly limited to what they -can- and -can not- do. This is a GOOD thing in the name of security. CHROOTing daemons like named isn't perfect but it does help.

- The "Split Zone" feature means that there will be (2) named processes running on your machine. One daemon will run and answer DNS queries for the *external* interface while the other daemon will answer on the *internal* interface for the private network. This setup helps protect your internal network IP addresses and names from being exposed to people out on the Internet. The more information you can hide from hostile users on the Internet, the harder it will be for them to break into your systems.

To setup your own domain, the first thing you need to do is get a domain from one of the Domain Registars listed at `<http://www.internic.net>`. There are lots of them out there and price and the quality of their services varies wildly. So far, I've had great luck with `<http://www.directnic.net>` since they offer the ability via an SSL encrypted WWW page vs. old-school mechanisms like email, etc. If you have questions about other registrars you're thinking of using, send me an email and I can give you my thoughts. Next, you need to find another DNS server out on the Internet that will be a SECONDARY dns server for your chosen Internet domain(s). This backup server is for the situations when your server or Internet connection goes down and you don't want to bounce email, etc. (see 5 (Section 24 - Sendmail) for more details about backup email services). Please note that getting this secondary server setup is NOT optional! Many domain registrars won't accept your domain name application without at least ONE backup domain server. Fortunately, many registrars can offer this secondary service for you for some additional fee. Again, prices vary wildly.

*** If you would like to read more on HOW to get your own domain names and understand some important legal issues with Internet domain names, please see the 24.21 (How to acquire a Domain Name) sub-section at the end of this section.**

24.1 Protecting your Internet Domain Name when Making Changes

- **NOTE: Due to the fact that DNS can make or break the Internet, you should be very sure that any updates, changes, etc. submitted to the Internic for your domain is done in a secure fashion. I personally recommend that you do all of your Internic updates to your registrar either via a SSL encrypted WWW page or via PGP encrypted email instead of the default old school "Mail-From" email method. Why? Email is *very* easy to forge. Because of this, it would be easy for a hostile user to screw up your domain name, take ownership of it, etc.**

  PGP and GnuPG for Linux will be covered in a future chapter but until then, I recommend to either use the Windows PGP client or at least use the Internic's "crypt-pw" option.

24.2 BIND version 9 vs 8 vs 4 and Figuring out what version you have:

This document is intended for BIND versons 9.1.x (and newer) as well as 8.3.x. If you are still running Bind4 or even Bind8, you really need to upgrade because you are either vunerable to ROOT hacks and/or these versions are old and are either soon to be or are already unsupported.

Just a little history:

- Bind 4.x was the defacto DNS server that helped start the Internet boom. It used the "named.boot" file and lived a long life. ISC then later overhauled BIND with version 8 which added lots of things including Dynamic DNS, IPv6, updated the zone file formats, and added a LOT of other features. With this new version, ISC changed the master configuration file to be "named.conf". With Bind 9, ISC has yet again done another major overhaul. This new version of Bind has added DNSSEC (signed DNS zones tranfered over encrypted SSL connections) as well as added direct database support (for MASSIVE zone files) vs. using the classic flat files as described here. Beyond that, the zone files stayed mostly the same between v8 and v9 except for minor formatting changes and the multitude of new optional features.

If you are unsure what version you have installed, you can find out the version from one of two ways.

- #1: If you have a LOCAL account on the DNS server, log into it and run one of the following commands:
  - "strings /home/chroot-dns-ext/usr/sbin/named | grep 9.2"
  - "strings /home/chroot-dns-ext/usr/sbin/named | grep 9.1"
  - "strings /home/chroot-dns-ext/usr/sbin/named | grep 9.0"
  - "strings /home/chroot-dns-ext/usr/sbin/named | grep 8. | grep REL"

  Or if it's not a CHROOTed DNS server:

  - "strings /usr/sbin/named | grep named"
  - "strings /usr/sbin/named | grep 9.2."
  - "strings /usr/sbin/named | grep 9.1."
  - "strings /usr/sbin/named | grep 9.0."
  - "strings /usr/sbin/named | grep 8. | grep REL"

  From the output, carefully look through the results until you find the version number. You will typically find it somewhere in the middle of the results for Bind 9.x and on the bottom for Bind 8.x.

- #2: If the DNS server is remote or you don't have an account to log into it, do the following on a local machine that has the **dig** or the older **nslookup** program: The new way using the `dig` (might not work on older version of Dig):
  - Run "`dig @ns1.xyz.com chaos txt version.bind`" from the command prompt where "ns1.xyz.com" is one of the DNS server(s) you are trying get the Bind version number from. You can get the names of the DNS servers running for a given domain by running the command "whois xyz.com". That should tell you the version of the DNS server.

  Older method using nslookup (deprecated - nslookup is going away. Use Dig):

  - Run `nslookup` from the command prompt
  - At the > prompt, type in `server xyz` (return) where `xyz` is the IP or name of the remote DNS server.
  - Now type in `set q=txt` (return) and then `set class=chaos` (return).
  - Finally, type in `version.bind` (return). That should tell you the version.
  - Hit Control-D to exit out of nslookup.

24.3 Security Warnings about previous versions of BIND

There are several MAJOR security exploits out there for older versions of Named (8.3.3-REL, 8.2.5, etc.). Make sure you are running at LEAST version **8.3.4, 9.2.2, or newer**. It should be noted that 9.2.2 requires a non-vulnerable version of OpenSSL to be installed if you want to use the "–with-openssl" feature. TrinityOS doesn't currently cover this topic but the installation of 9.2.2 is highly recommended. If you aren't running the newest code, you will be vulnerable to hostile users getting ROOT access on your box!

**** To stay up on the newest Bind releases, I recommend that ALL users add themselves to the BIND-announce email list given in 5 (Section 5).**

This email list is ONLY for BIND version announcements and is very low on email traffic.

24.4 Downloading and compiling BIND

- First, download ISC's "named" server code from the URL in 5 (Section 5) and put it into a directory such as */usr/src/archive/bind/*. It is also highly recommend to download ISC's PGP key so you can verify that this code hasn't be altered by any hostile users. You should also check and download any required patches, etc. if there are any.
- Next, go into that new directory and both VERIFY and uncompress the archive
- **Bind 9.2.x** specific instructions:
  -

  -
    ```
    cd /usr/src/archive/bind/

    #Assuming you have GPG installed (but not nessisarily configured), you will
    #need to download both ISC's PGP key and the .asc PGP signature file for the
    #Bind source code.  Please note that ISC seems to keep changing their PGP keys
    #from time to time so your current ISC key might be old now.  So let's verify
    #that the code is legit:
    #
    # replace x.y.z with the correct version of Bind you are using
    #
          gpg --import pgpkey2004.txt
          gpg --verify bind-9.2.3.tar.gz.asc

    #Make sure it says "Good Signature" at the top. There might be some trust
    # warnings but don't worry about that.
    ```

  -
    ```
    #So if the above PGP section passed (or you skipped it), now do the
    #  following:
    #
    #The Bind 9 archive creates its own subdirectory so there is no need to
    #  create one
    #

          tar xzvf bind-9.x.y.tar.gz
    ```
- **Bind 8.3.x** specific instructions:

  -
    ```
    #I haven't added PGP verification for Bind 8.x as it's old and you
    #  really should install Bind9.  Anyway, for those of you who want Bind8:
    #
    #The Bind 8 archive does NOT create its own subdirectory so I recommend
    to create one first
    #
          mkdir /usr/src/archive/bind/8.x.y
          mv /usr/src/archive/bind* /usr/src/archive/bind/8.x.y
          cd /usr/src/archive/bind/8.x.y
          tar xzvf bind-src.tar.gz
          tar xzvf bind-doc.tar.gz
    ```
- **Bind 9.x.y specific compiling:**

- Go into that new directory and run the *configure* script

```
cd /usr/src/archive/bind/bind-9.x.y

# For Bind 9.2.x
# ----------------
# The various compiling configurations are now configured via Automake
#
# Not only that but ISC has again changed their paths and such.  So,
# the following setup will place files into their more "classic"
# directories
#
# Please note the "--disabled-threads" option.
#
#   This tag will allow CHROOT DNS to work under Linux 2.2.x kernels.
#   The reason for this is that there is a bug in ALL 2.2.x kernels
#   that basically makes CHROOTing things broken BUT it was fixed
#   in the 2.4.x kernels.  If you are running a 2.4.x kernel, you do
#   NOT need this option.  See the end of the "named" MAN page
#   for more details about this.
#
# Please note that the "--exec-prefix" stuff on the ./configure line
# will put BIND into the /usr/sbin directory (the default is /usr/local
# (bin, sbin, etc.)) which is the stock place for Mandrake.  You can
# put these binaries as well as documentation anywhere you wish.  If
# you would like to put it in the proper place for your distribution,
# run the command:
#
#         whereis named
#
# to find out where they put the binaries and such and then substitute
# this new path for the Automake one above.  REMEMBER this path for
# later in this section when making the CHROOT jails!
#
#----------------------------------------------------------------------

#2.4.x kernels only
#
./configure --prefix= --exec-prefix=/usr --datadir=/usr/share \
--includedir=/usr/include --infodir=/usr/share/info \
--mandir=/usr/share/man


#2.2.x kernels only
#
./configure --prefix= --exec-prefix=/usr --datadir=/usr/share \
--includedir=/usr/include --infodir=/usr/share/info \
--mandir=/usr/share/man --disable-threads

#All kernels - 2.4 or 2.x
#
```

```
make
```

  - From here, the machine should compile things up without any issues. Compile times will vary depending on the speed and available resources on your machine.

- **Bind 8.3.x Specific compiling:** Go into that new directory and compile things up

  -
    ```
    cd /usr/src/archive/bind/8.3.4/src

    # For Bind 8.3.4
    # ----------------
    # The various compiling configurations are now configured in the
      # port/linux/Makefile.set file.
    #
    # Interestingly enough, ISC has now made /usr/sbin/ the default directory
    # so you shouldn't have to do anything special beyond that
    #
    # Note:
    # -----
    # FYI, Bind 8.2.4 would NOT compile on my Mandrake 2.2.19 machine as
    # it would give me the following error:
    #
    #  eventlib.c:296: structure has no member named `fds_bits' . . .
    #
    # To fix this, edit the file "src/port/linux/include/port_before.h" and
    # insert the following line after the existing "define" lines:
    #
    #    #define _GNU_SOURCE
    #
    # Ok, before you try to compile the code up again, run the command
    #  "make clean"
    #
    # ----------------------------------------------------------------------

    #Ok.. compile it up

    make clean
    make all
    ```

  - From here, the machine should compile things up without any issues. Compile times will vary depending on the speed and available resources on your machine.

- **Final installation steps for ALL versions of BIND:**

  - Once the compiling is finished, install your new version of Bind by running:

```
make install
```

  - **For Bind9 users:** Starting with Bind 9.x, ISC creates the MAN and HTML documentation files via SGML. Unless you have the OpenJade program installed on the machine and it's in the path, the "make install" process will quietly mention that OpenJade wasn't found and thus the docs won't be created and installed. Fortunately, ISC has pre-built MAN files in there though unfortunately they DON'T use these during the "make install" phase. Here is a work around:

```
cd /usr/src/archive/bind/bind-9.2.x
find . -name "*.1" -exec cp {} /usr/share/man/man1/ \;
find . -name "*.3" -exec cp {} /usr/share/man/man3/ \;
find . -name "*.5" -exec cp {} /usr/share/man/man5/ \;
find . -name "*.8" -exec cp {} /usr/share/man/man8/ \;

#you could have also done it with xargs too:
#find . -name "*.1" | xargs -i cp {} tmp
```

- **For Bind8 users:** ISC no longer includes the installation of the documentation within the Makefile so lets move them over manually:

```
cd /usr/src/archive/bind/bind-8.3.4/doc/man
make clean
make all
make install
```

24.5 Creating the CHROOTed environments

Now, follow the procedures to create the required chrooted user login, group, and various files and do any required substitutions where required.

- First, create the "chroot-dns-ext" user group for the CHROOTed EXTERNAL interface:

```
groupadd -g 120 chroot-dns-ext
```

- Next, create the "chroot-dns-int" group for the CHROOTed INTERNAL interface:

```
groupadd -g 121 chroot-dns-int
```

- Now create the "chroot-dns-ext" and "chroot-dns-int" user for the CHROOTed EXTERNAL and INTERNAL interfaces:

```
useradd -u 120 -g 120 chroot-dns-ext
useradd -u 121 -g 121 chroot-dns-int
```

- The next steps is to create the actual various chroot'ed directories, fix their permissions, etc:

```
#  Since this is a CHROOTed environment, you need to make this little
#  world look like the real one.  This means you need the required
#  system directorys as well.

     cd /home/chroot-dns-ext
```

```
mkdir -p etc lib dev usr/sbin var/named var/run
chmod -R 750 /home/chroot-dns-ext
mknod -m 666 dev/null c 1 3
mknod -m 666 dev/zero c 1 5
mknod -m 666 dev/random c 1 8

cd /home/chroot-dns-int

mkdir -p etc lib dev usr/sbin var/named var/run
chmod -R 750 /home/chroot-dns-int
mknod -m 666 dev/null c 1 3
mknod -m 666 dev/zero c 1 5
mknod -m 666 dev/random c 1 8
```

- Now, we need to copy over the required libraries and executable files.
  - NOTE: Whenever you patch your machine and some of the patches include updated GLIBC files, you will need to REPEAT this section to put a new copy of the updated libraries into the various CHROOT directories.

```
cp -f /lib/libc.so.6 /home/chroot-dns-ext/lib
cp -f /lib/libc.so.6 /home/chroot-dns-int/lib
cp -f /lib/ld-linux.so.2 /home/chroot-dns-ext/lib
cp -f /lib/ld-linux.so.2 /home/chroot-dns-int/lib
```

****NOTE: You will notice that I recommend to first COPY and then later MOVE the executables into the CHROOT'ed directory. This gives you a little more slack in case you make a mistake before you finally remove the original files.**

```
cp -f /usr/sbin/named* /home/chroot-dns-ext/usr/sbin
chmod 750 /home/chroot-dns-ext/usr/sbin/named*
mv -f /usr/sbin/named* /home/chroot-dns-int/usr/sbin
chmod 750 /home/chroot-dns-int/usr/sbin/named*
```

Ok, fix the binary's file owner and group permissions:

```
chown -R chroot-dns-int.chroot-dns-int /home/chroot-dns-int
chown -R chroot-dns-ext.chroot-dns-ext /home/chroot-dns-ext
```

24.6 Creating the internal named.conf configuration file

- Ok, time to create the actual DNS Zone files. These are the full authoritative configs for both Bind 9.x.y as well as Bind v8.x.y:

NOTE: You'll notice that some lines will SEEM to have extra "."s (periods) at the end of domain names, etc. LEAVE THEM THERE!! They are supposed to be there and are CRITICAL to bind's internal file format!

/home/chroot-dns-int/etc/named.conf

```
// /home/chroot-dns-int/etc/named.conf for TrinityOS - 01/12/03

// Config file for a full authoritative --INTERNAL-- DNS server
//
//  This internal server will be the one use by the DNS server itself
//  and by any internal hosts as well

options {
        //Remember, this is already CHROOTed.  /var/named IS correct
                directory "/var/named";

        //You dont want the external interface to listen on this zone
                listen-on port 53 {
                192.168.0.1; 127.0.0.1;
                };

        // Uncommenting this might help if you have to go through a
        // firewall and things are not working out:
        // query-source address * port 53;
};


// Filter out any LAME server messages from cluttering up the SYSLOGs
        logging {
         category "lame-servers" { null; };
        };


zone "." {
        type hint;
        file "root.hints.db";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        notify no;
        file "127.0.0.db";
};

zone "acme123.com" {
        type master;
        notify no;
        file "acme123-int.com.db";
        allow-transfer { none; };
        allow-query { 127/8; 192.168.0/24; };
};

zone "0.168.192.in-addr.arpa" {
        type master;
        notify no;
```

```
        file "192.168.0-in.addr.db";
        allow-transfer {none; };
        allow-query {127/8; 192.168.0/24; };
};
```

You will notice that I am filtering out *LAME SERVER* messages from being sent to SYSLOG. What is a "lame server"?

- Basically, when you try to resolve some domain name that is specified from the Internic (as found in "whois") as an AUTHORITATIVE DNS server, the server should reply to that original DNS request as "authoritative". A LAME server is a DNS server that doesn't reply back with an authoritative answer but with a "non-authoritative" answer instead. Basically, DNS requests will still work but you now know that the remote DNS server is mis-configured. So why should you filter these messages? First, there is nothing you can do about these messages other than emailing EVERY remote domain and telling that their server is broken. There are a LOT of LAME servers out on the Internet and all these warning errors will fill up your logs quickly. So I say screw it, let them fix their mess, and until they do, stop logging all this.

24.7 Creating the internal zone files

- Next, you need to create the root.hints.db file like the one shown below. Basically, this file tells your DNS server how to reach the multiple Internet ROOT servers. But, like anything else, the IP address of the various root servers are always changing. So, I recommend you create your OWN root.hints.db file by running the following command and not using the below example .db file:

```
dig @a.root-servers.net . ns > /home/chroot-dns-int/var/named/root.hints.db
```

/home/chroot-dns-int/var/named/root.hints.db

```
; <<>> DiG 8.1 <<>> @a.root-servers.net . ns
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.                       5d10h28m15s IN NS  M.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  L.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  K.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  J.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  B.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  F.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  G.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  C.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  H.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  A.ROOT-SERVERS.NET.
```

```
.                       5d10h28m15s IN NS  D.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  E.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  I.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17

;; Total query time: 15115 msec
;; FROM: ns.acme123.com to SERVER: a.root-servers.net 198.41.0.4
;; WHEN: Fri Oct  1 03:02:15 1999
;; MSG SIZE  sent: 17  rcvd: 436
```

The following file is the REVERSE zone records for the "localhost" or loopback interface:

/home/chroot-dns-int/var/named/127.0.0.db

```
;
; /home/chroot-dns-int/var/named/127.0.0.db ZONE file for TrinityOS - 09/03/01
;
$TTL    86400
@              IN      SOA     ns.acme123.com. hostmaster.acme123.com. (
                            2001052800      ; serial, todays date + todays serial #
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL

                          NS      ns.acme123.com.

1                 86400  PTR     localhost.acme123.com.
```

The following file is the FORWARD zone record for the internal ACME123.com network

/home/chroot-dns-int/var/named/acme123-int.com.db

```
;
; /home/chroot-dns-int/var/named/acme123-int.com ZONE file for TrinityOS - 09/03/01
;
$TTL    86400
@       IN      SOA     ns.acme123.com. hostmaster.acme123.com. (
```

```
                                2001052800      ; serial, todays date + todays serial #
                                8H              ; refresh, seconds
                                2H              ; retry, seconds
                                1W              ; expire, seconds
                                1D )            ; minimum, seconds

                                NS      ns.acme123.com.     ; Inet Address of name server
                                NS      ns.backupacme.com.  ; Inet address of backup server
                                MX      10  mail.acme123.com.   ; Primary MX server

;
; note - If you wish to directly resolve any acme123.com hosts
;        that are currently only defined in the EXTERNAL zone
;        files (say www.acme123.com), you MUST list them here
;        as well since the internal zone assumes that it is
;        authoritative for acme123.com zone and thus would never
;        contact the external server for any other
;        acme123.com queries.


roadrunner-int      86400       A       192.168.0.1
                                HINFO   "a486/160/40M" "Linux 2.0"

mail                86400       A       192.168.0.1
                                HINFO   "a486/160/40M" "Linux 2.0"


coyote              86400       A       192.168.0.2
                        HINFO   "iPentium-II/260/64M"  "Win95"

spare               86400       A       192.168.0.9
                        HINFO   "Unknown" "Unknown"

spare2              86400       A       192.168.0.10
                        HINFO   "Unknown" "Unknown"
```

The following file is the REVERSE zone record for the internal ACME123.com network

/home/chroot-dns-int/var/named/192.168.0-in.addr.db

```
;
; /home/chroot-dns-int/var/named/192.168.0-in.addr ZONE file for TrinityOS - 09/03/01
;
$TTL    86400
@               IN      SOA     ns.acme123.com. hostmaster.acme123.com. (
                            2001052800      ; serial, todays date + todays serial #
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
```

```
                          NS      ns.acme123.com.

1                       86400   PTR     roadrunner-int.acme123.com.
2                       86400   PTR     coyote.acme123.com.

9                       86400   PTR     spare.acme123.com.
10                      86400   PTR     spare2.acme123.com.
```

24.8 Creating the external named.conf configuration file

- Now, here is the configuration file for the EXTERNAL DNS server:

/home/chroot-dns-ext/etc/named.conf

```
// /home/chroot-dns-ext/etc/named.conf for TrinityOS - 11/25/02
// Config file for a full authoritative --EXTERNAL-- DNS server

options {
    //Remember, this is already CHROOTed.  /var/named IS correct
    directory "/var/named";

    //Do NOT have the server listening on localhost or the internal interface
    listen-on port 53 {
      100.200.0.212;
    };

    // Clean the cache every 6 hours (default is 1).
    cleaning-interval 360;

    // Do NOT respond to DNS queries for any domains other than local zones
    //
    //   All remote DNS lookups for this host and any internal machines will
    //   be served from the INTERNAL DNS server
    recursion no;

    // Uncommenting this might help if you have to go through a
    // firewall and things are not working out:
   // query-source address * port 53;
};

zone "." {
        type hint;
        file "root.hints.db";
};

zone "acme123.com" {
        type master;
        notify yes;
        file "acme123.com.db";
```

```
        allow-transfer {
           102.200.0.25/32;
        };
};

zone "212.0.200.100.in-addr.arpa" {
        type master;
        notify yes;
        file "212.0.200.100.db";
    allow-transfer {
      102.200.0.25/32;
        };
};
```

24.9 Creating the external zone files

- Next, you need to create another root.hints.db file like the one shown below. But, like any thing else, the Internet's root servers are always changing. So, I recommend you create your OWN copy by running the following command and not using the below example .db file:

```
                dig @a.root-servers.net . ns > /home/chroot-dns-ext/var/named/root.hints.db
```

/home/chroot-dns-ext/var/named/root.hints.db

```
; <<>> DiG 8.1 <<>> @a.root-servers.net . ns
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.                       5d10h28m15s IN NS  M.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  L.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  K.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  J.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  B.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  F.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  G.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  C.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  H.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  A.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  D.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  E.ROOT-SERVERS.NET.
.                       5d10h28m15s IN NS  I.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
```

```
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17

;; Total query time: 15115 msec
;; FROM: ns.acme123.com to SERVER: a.root-servers.net 198.41.0.4
;; WHEN: Fri Oct  1 03:02:15 1999
;; MSG SIZE  sent: 17  rcvd: 436
```

The following file is the FORWARD zone records for the external ACME123.com network

/home/chroot-dns-ext/var/named/acme123.com.db

```
;
; /home/chroot-dns-ext/var/named/acme123.com ZONE file for TrinityOS - 09/03/01
;
$TTL    86400
@       IN      SOA     ns.acme123.com. hostmaster.acme123.com. (
                                2001052800      ; serial, todays date + todays serial #
                                8H              ; refresh, seconds
                                2H              ; retry, seconds
                                1W              ; expire, seconds
                                1D )            ; minimum, seconds

               NS      ns.acme123.com.      ; Inet Address of name server
               NS      ns.backupacme.com.   ; Inet address of backup server

               MX   10 mail.acme123.com.    ; Primary Mail Exchanger


ns              86400   A       100.200.0.212
                                HINFO   "a486/160/40M" "Linux 2.0"

mail            86400   A       100.200.0.212
                                HINFO   "a486/160/40M" "Linux 2.0"


ftp             86400   CNAME   ns

roadrunner      86400   CNAME   ns
```

The following file is the REVERSE zone records for the external ACME123.com network:

/home/chroot-dns-ext/var/named/212.0.200.100.db

```
;
; /home/chroot-dns-ext/var/named/212.0.200.100-in.addr ZONE file for TrinityOS - 09/03/01
;
$TTL    86400
@       IN      SOA     ns.acme123.com. hostmaster.acme123.com. (
                    2001052800      ; serial, todays date + todays serial #
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL

                    NS      ns.acme123.com.    ; Inet Address of name server
                    NS      ns.backupacme.com. ; Inet address of backup server

212.0.200.100.in-addr.arpa. IN PTR     ns.acme123.com.
```

24.10 Fixing final CHROOTed permissions and ownerships

- Ok, lets finally fix the file owner and group permissions for the respective Zone files:

```
        chown -R chroot-dns-int.chroot-dns-int /home/chroot-dns-int
        chown -R chroot-dns-ext.chroot-dns-ext /home/chroot-dns-ext
```

24.11 Tuning How NAMED loads the SPLIT zone file configuration

Ok, time for the glue. You need to change the way that DNS loads the server up to recognize the new CHROOT layout and to load the SPLIT servers:

Redhat users:

- Edit /etc/rc.d/init.d/named and change the lines:

```
                        [ -f /usr/sbin/named ] || exit 0
                        .
                        .
                        .
                        [ -f /etc/named.conf ] || exit 0
```

to:

```
                        [ -f /home/chroot-dns-int/usr/sbin/named ] || exit 0
                        [ -f /home/chroot-dns-ext/usr/sbin/named ] || exit 0

                        [ -f /home/chroot-dns-int/etc/named.conf ] || exit 0
                        [ -f /home/chroot-dns-ext/etc/named.conf ] || exit 0
```

- You now need to setup the following lines to do the actually loading of the two individual DNS servers. It is recommented that you get this file from the TrinityOS-security script at `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>` to save you time and avoid possible typos.
- It's IMPORTANT that you edit this file and enable the correct version of Bind that you plan on running. To disable a specific version, place "#" charecters in the front of the respective lines.

```
#!/bin/sh
#
# named           This shell script takes care of starting and stopping
#                 named (BIND DNS server).
#
# chkconfig: - 55 45
# description: named (BIND) is a Domain Name Server (DNS) \
# that is used to resolve host names to IP addresses.
# probe: true



# ----------------------------------------------------------------------------
# # TrinityOS-named
# v11/25/02
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/~dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
#
#  NOTE: It's IMPORTANT that you edit this file and enable the correct
#        version of Bind that you plan on running.  To disable a specific
#        version, place "#" charecters in the front of the respective lines.
#
#        Bind9 is the TrinityOS default setting.
#
#
# Updates
# -------
# 11/25/02 - Updated some of the comments
#
# 03/05/01 - Updated the file to support the loading of Bind9
#
# 01/28/01 - Added a few CR-LFs to clean up the output between starting
#            the internal and external zones
# 10/07/00 - Added the start-int, start-ext, stop-int, and stop-ext functions
#
# ----------------------------------------------------------------------------


# Source function library.
```

```
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /home/chroot-dns-int/usr/sbin/named ] || exit 0
[ -f /home/chroot-dns-ext/usr/sbin/named ] || exit 0

[ -f /home/chroot-dns-int/etc/named.conf ] || exit 0
[ -f /home/chroot-dns-ext/etc/named.conf ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in

       start)
            # Start daemons.
            echo -n "Starting named-int: "

            #Bind9 - Use this setup if you are using Bind9
            #
            daemon /home/chroot-dns-int/usr/sbin/named -u chroot-dns-int -t /home/chroot-dns-int

            #Bind8 - # out the "daemon" line above and un-# out the line below
            #        if you are running Bind8
            #
            #daemon /home/chroot-dns-int/usr/sbin/named -u chroot-dns-int -g chroot-dns-int -t /home/chroot-dns-i

            RETVAL=$?
            [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named-int

            sleep 5

            echo -e "\r"
            echo -n "Starting named-ext: "

            #For some reason, this server won't load with the "daemon" line in
            # front - if you have a solution for this, please let me know

            #Bind9 - Use this setup if you are using Bind9
            #
            /home/chroot-dns-ext/usr/sbin/named -u chroot-dns-ext -t /home/chroot-dns-ext

            #Bind8 - # out the "daemon" line above and un-# out the line below
            #        if you are running Bind8
            #
```

```
    #/home/chroot-dns-ext/usr/sbin/named -u chroot-dns-ext -g chroot-dns-ext -t /home/chroot-dns-ext

    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named-ext
    echo -e "\r"
 ;;

start-int)
    # Start daemons.
    echo -n "Starting named-int: "

    #For some reason, this server won't load with the "daemon" line in
    # front - if you have a solution for this, please let me know

    #Bind9 - Use this setup if you are using Bind9
    #
    /home/chroot-dns-int/usr/sbin/named -u chroot-dns-int -t /home/chroot-dns-int

    #Bind8 - # out the "daemon" line above and un-# out the line below
    #        if you are running Bind8
    #
    #/home/chroot-dns-int/usr/sbin/named -u chroot-dns-int -g chroot-dns-int -t /home/chroot-dns-int

    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named-int
    echo -e "\r"
;;

start-ext)
    echo -n "Starting named-ext: "

    #For some reason, this server won't load with the "daemon" line in
    # front - if you have a solution for this, please let me know

    #Bind9 - Use this setup if you are using Bind9
    #
    /home/chroot-dns-ext/usr/sbin/named -u chroot-dns-ext -t /home/chroot-dns-ext

    #Bind8 - # out the "daemon" line above and un-# out the line below
    #        if you are running Bind8
    #
    /home/chroot-dns-ext/usr/sbin/named -u chroot-dns-ext -g chroot-dns-ext -t /home/chroot-dns-ext

    RETVAL=$?
    $RETVAL -eq 0 ] && touch /var/lock/subsys/named-ext
    echo -e "\r"
;;

stop)
  # Stop daemons.
```

```
          echo -n "Shutting down named: "
   killproc named
   RETVAL=$?
   [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named-int && rm -f /var/lock/subsys/named-ext
   echo -e "\r"
;;

stop-int)
    # Stop INT daemons.
    echo -n "Shutting down named-int: "
    kill `ps ax | grep chroot-dns-int/usr/sbin/named | grep -v -e grep | awk '{print $1}'`
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named-int
    echo -e "\r"
;;

stop-ext)
    # Stop EXT daemons.
    echo -n "Shutting down named-ext: "
    kill `ps ax | grep chroot-dns-ext/usr/sbin/named | grep -v -e grep | awk '{print $1}'`
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named-ext
    echo -e "\r"
;;

status)
  /usr/sbin/ndc status
  exit $?
;;


restart)
   $0 stop
   $0 start
;;


reload)
   /usr/sbin/ndc reload
   exit $?
;;


probe)
   # named knows how to reload intelligently; we don't want linuxconf
   # to offer to restart every time
   /usr/sbin/ndc reload >/dev/null 2>&1 || echo start
   exit 0
;;
```

```
        *)
          echo "Usage: named {start|start-int|start-ext|stop|stop-int|stop-ext|status|restart}"
        exit 1
esac
exit $RETVAL
```

24.12 Fixing SYSLOGing to understand the new CHROOTed setup

- Next, we need now modify how SYSLOG loads up so it understands how to deal with the new CHROOTed DNS servers: Edit the `/etc/rc.d/init.d/syslog` file and change the loading of SYSLOG to the following:

```
        daemon syslogd -a /home/chroot-dns-int/dev/log -a /home/chroot-dns-ext/dev/log -m 0
```

Now, configure your machine to use the local DNS server by editing /etc/resolv.conf

- This is a CRITICAL setting. If you configure the Linux machine to use the EXTERNAL IP address or one of your ISP's DNS server, the Linux server won't be able to resolve any of the Internet hosts due to the SPLIT server setup.

```
        search acme123.com
        nameserver 127.0.0.1

        #Backup - your ISP's DNS servers
        #nameserver 10.200.200.69
        #nameserver 10.200.200.96
```

Next, make sure that your machine is prepped to use DNS:

Slackware: /etc/host.conf

```
                order hosts, bind
                multi on
```

Redhat: /etc/nsswitch.conf

Change the "hosts" line to read:

```
                                "hosts:      files dns"
```

Also, I would recommend to DELETE all instances of NIS from each line of this file UNLESS you *ARE* using NIS, NIS+, etc!

24.13 Starting up and testing BIND

Ok, getting close! Now, make sure that BIND is enabled to load upon boot.

- To do this, UN-DO all edits done to disable DNS in 8 (Section 8)
  - **Note:** the NTSYSV method won't work for all of this
- Now, test that all the named files are correct by running "named" in a forground test (For Bind9):

```
/home/chroot-dns-int/usr/sbin/named -u chroot-dns-int -t /home/chroot-dns-int -f
```

The **INTERNAL** server output should look something like this for Bind 9.2.x:

```
Nov 25 22:34:01 roadrunner named[1959]: starting BIND 9.2.1 -u chroot-dns-int -t /home/chroot-dns-int
Nov 25 22:34:01 roadrunner named[1959]: using 1 CPU
Nov 25 22:34:02 roadrunner named[1959]: loading configuration from '/etc/named.conf'
Nov 25 22:34:02 roadrunner named[1959]: no IPv6 interfaces found
Nov 25 22:34:02 roadrunner named[1959]: listening on IPv4 interface lo, 127.0.0.1#53
Nov 25 22:34:02 roadrunner named[1959]: listening on IPv4 interface eth1, 192.168.0.1#53
Nov 25 22:34:02 roadrunner named[1959]: listening on IPv4 interface eth2, 192.168.10.1#53
Nov 25 22:34:02 roadrunner named[1959]: zone 0.0.127.in-addr.arpa/IN: loaded serial 2001022400
Nov 25 22:34:02 roadrunner named[1959]: zone 0.168.192.in-addr.arpa/IN: loaded serial 2002102600
Nov 25 22:34:02 roadrunner named[1959]: zone 10.168.192.in-addr.arpa/IN: loaded serial 2001031101
Nov 25 22:34:02 roadrunner named[1959]: zone acme123.com/IN: loaded serial 2002112500
Nov 25 22:34:02 roadrunner named[1959]: running
```

The **INTERNAL** server output should look something like this for Bind 8.3.x:

```
Apr 10 01:48:42 roadrunner named[27951]: starting.  named 8.3.4 Tue Dec 14 20:30:23 CET 1999 ^Iroot@jedi.man
/RPM/BUILD/bind-8.2.2P5/src/bin/named
Apr 10 01:48:42 roadrunner named[27951]: hint zone "" (IN) loaded (serial 0)
Apr 10 01:48:42 roadrunner named[27951]: Zone "192.168.0" (file 192.168.0.db): No default TTL set using SOA
Apr 10 01:48:42 roadrunner named[27951]: master zone "192.168.0" (IN) loaded (serial 2000033100)
Apr 10 01:48:42 roadrunner named[27951]: Zone "0.168.192.in-addr.arpa" (file 192.168.0-in.addr.db): No defaul
minimum instead
Apr 10 01:48:42 roadrunner named[27951]: master zone "0.168.192.in-addr.arpa" (IN) loaded (serial 1999111300
Apr 10 01:48:42 roadrunner named[27951]: listening on [127.0.0.1].53 (lo)
Apr 10 01:48:42 roadrunner named[27951]: Forwarding source address is [0.0.0.0].1033
Apr 10 01:48:42 roadrunner named[27951]: chrooted to /home/chroot-dns-int
Apr 10 01:48:42 roadrunner named[27951]: group = chroot-dns-int
Apr 10 01:48:42 roadrunner named[27951]: user = chroot-dns-int
Apr 10 01:48:42 roadrunner named[27951]: Ready to answer queries.
Apr 10 01:48:42 roadrunner named[27951]: Zone "192.168.0" (file 192.168.0.db): No default TTL set using SOA
Apr 10 01:48:42 roadrunner named[27951]: Zone "0.168.192.in-addr.arpa" (file 192.168.0-in.addr.db): No defaul
minimum instead
```

Hit Control-C when you are sure that Named is running ok and it's running the correct version of Named.

Now try running the external server:

```
/home/chroot-dns-ext/usr/sbin/named -u chroot-dns-ext -g chroot-dns-ext -t /home/chroot-dns-ext
```

The **EXTERNAL** server output should look something like this for Bind 9.2.x:

```
Nov 25 22:34:07 roadrunner named[1965]: starting BIND 9.2.1 -u chroot-dns-ext -t /home/chroot-dns-ext
Nov 25 22:34:07 roadrunner named[1965]: using 1 CPU
Nov 25 22:34:07 roadrunner named[1965]: loading configuration from '/etc/named.conf'
Nov 25 22:34:07 roadrunner named[1965]: no IPv6 interfaces found
Nov 25 22:34:07 roadrunner named[1965]: listening on IPv4 interface eth0, 64.220.150.140#53
Nov 25 22:34:07 roadrunner named[1965]: zone 212.0.200.100.in-addr.arpa/IN: loaded serial 2002070700
Nov 25 22:34:07 roadrunner named[1965]: zone acme123.com/IN: loaded serial 2002070700
Nov 25 22:34:07 roadrunner named[1965]: running
```

The **EXTERNAL** server output should look something like this for Bind 8.3.x:

```
Apr 10 01:52:10 roadrunner named[27960]: starting.  named 8.3.4 Tue Dec 14 20:30:23 CET 1999 ^Iroot@jedi.man
RPM/BUILD/bind-8.2.2P5/src/bin/named
Apr 10 01:52:10 roadrunner named[27960]: hint zone "" (IN) loaded (serial 0)
Apr 10 01:52:10 roadrunner named[27960]: Zone "acme123.com" (file acme123.com.db): No default TTL set using
Apr 10 01:52:10 roadrunner named[27960]: master zone "acme123.com" (IN) loaded (serial 2000033100)
Apr 10 01:52:10 roadrunner named[27960]: Zone "212.0.200.100.in-addr.arpa" (file 100.200.0.212.db): No defaul
minimum instead
Apr 10 01:52:10 roadrunner named[27960]: master zone "212.0.200.100.db.in-addr.arpa" (IN) loaded (serial 200
Apr 10 01:52:10 roadrunner named[27960]: listening on [100.200.0.212].53 (eth0)
Apr 10 01:52:10 roadrunner named[27960]: Forwarding source address is [0.0.0.0].1033
Apr 10 01:52:10 roadrunner named[27961]: chrooted to /home/chroot-dns-ext
Apr 10 01:52:10 roadrunner named[27961]: group = chroot-dns-ext
Apr 10 01:52:10 roadrunner named[27961]: user = chroot-dns-ext
Apr 10 01:52:10 roadrunner named[27961]: Ready to answer queries.
```

Hit Control-C when you are sure that NAMED is running ok and it's running the correct version of NAMED.

Please also note that if the TIME and DATE of your log files is off, you need to set the TZ environment variable as described in 7 (Section 7).

24.14 Possible Bind errors upon load

- **modprobe: can't locate module net-pf-10**
  named: no IPv6 interfaces found

  This error is due to Bind9 supporting IPv6 packets but your system doesn't. It sure would be nice if you could compile BIND without IPv6 support but you can't. To work around this, add the following to the /etc/module.conf (/etc/conf.module file for older distros):

```
        alias net-pf-10 off
```

- **named: none:0: open: /etc/rndc.key: file not found**
  named: couldn't add command channel 127.0.0.1#953: file not found

  The "rndc" program is a tool to manage local and remote named servers. It allows you to start / stop the server, increase debugging, reload the zone files, get stats, etc. TrinityOS doesn't cover the configuration or use of ndc/rndc because I've found using the /etc/rc.d/init.d/named tool just as good IMHO. Yes, it might create a minor lapse in service as you "restart" named but it's very minor.

- **named: could not open entropy source /dev/random: file not found**: You forgot to create a /dev/random in the CHROOT environment. Look above in this section for the "mknod" commands.

24.15 Enabling Bind to load upon future boots

- Now, do the following for your respective Linux Distribution:
  - Slackware Specific:
    - Un #'d out the lines in the "/etc/rc.d/rc.inet2" file for "named"
  - Redhat Specific:
    - Run the command "chkconfig –level=345 named on". Then make sure that the file "/etc/rc.d/rc3.d/S55named" exists

24.16 Changes for Bind9

As I mentioned before, TrinityOS currently doesn't currently cover advanced topics like Dynamic DNS, DNSSEC, etc. Some of these features are very cool and they WILL be covered some time in the future.

Anyway, for now, I wanted to mention that the "nslookup" that we are all familiar with is going away in favor of the "dig" and "host" commands instead. I recommend that you start getting used to using the "dig" and "host" commands. If you need to continue to use "nslookup", you should consider the following alias to avoid the annoying nslookup warnings:

/etc/bashrc

```
alias nslookup='nslookup -silent'
```

24.17 Supporting more than one Internet Domain name on this DNS server

Having your Linux box do DNS for more that just ONE domain is VERY simple. If you want to do this, all you have to do is:

1. Create another FORWARD zone file (e.g. another-domain.com) for your new domain. e.g. use the old acme123.com files from above as a template for your new /home/chroot-dns-ext/var/named/another-domain.com.db file
2. Edit the /home/chroot-dns-ext/etc/named.conf file to:
   (a) Add the loading of the new /var/named/newdomain.com.db zone file just like you did for the acme123.com zone file.
   (b) Allow your remote secondary DNS servers to be able access this new domain's zone file
   (c) Restart Bind

24.18 Setting up Secondary (BACKUP) DNS servers

If you want someone else's DNS server to be a secondary DNS server for your domain(s) *OR* you want your DNS server to be a secondary for someone else's domain(s), follow these following steps.

- Edit /home/chroot-dns-ext/etc/named.conf file and make sure the "allow-transfer" line has the proper IP address of the remote secondary DNS server. You can have as many secondary DNS servers as you want.
- Edit either your server's (if you want to be backup for some remote server) or the remote server's (if they are going to be a backup for your domains) `/home/chroot-dns-ext/etc/named.conf` file and APPEND the following:

```
zone "acme123.com." {
  type slave;
  file "acme123.com.db";
  masters { 100.200.0.212; };
  allow-transfer { none; };
};

zone "212.0.200.100.in-addr.arpa." {
  type slave;
  file "212.0.200.100.db";
  masters { 100.200.0.212; };
  allow-transfer { none; };
};
```

NOTE: If the remote domain actually had multiple IPs or a "subnet of IPs" (typically 5 or more IP addresses), you would need a slightly different configuration. The following example would be correct if the remote domain had -8- IP allocated.

```
zone "128/29.0.200.100.in-addr.arpa." {
  type slave;
  file "128.0.200.100.db";
  masters { 100.200.0.129; };
  allow-transfer { none; };
};
```

Basically, you need to understand that:

The IP addresses the remote site was given an address range of 100.200.0.128 through .135 with a subnet mask of 255.255.255.248 (a /29).

Then, with the not-so-obvious DNS syntax from RFC 2317, you read the top line as:

- In the last octet of the IP address, the first IP address of this remote subet is "128". (This is the NETWORK address)
- Next, the subnet mask is a /29 or 8 IPs
- The remaining reverse zone is 0.200.100

Yes, its weird syntax and NOT obvious (try even reading the RFC!) but it works fine.

- Finally, you need to create a dummy file for this remote domain.

```
touch /home/chroot-dns-ext/var/named/acme123.com.db
```

- Now, restart the remote secondary DNS server by running the following command from the remote box:
  - Redhat:

```
/etc/rc.d/init.d/named stop
/etc/rc.d/init.d/named start
```

  - Slackware:

```
kill -HUP ‘ps aux | grep named | grep -v -e grep | awk ’{print $2}’‘/usr/sbin/named &
```

Once everything is working fine, be SURE to follow the "aliases" instruction in 18 (Section 18).

24.19 Gotchas with Master DNS servers being down for long periods of time

IMPORTANT:

- If the MASTER DNS server for a given domain(s) is either down or unavailable for more than (1) week, that domain will then become unavailable and unresponsive REGARDLESS if there are any other secondary DNS servers for that domain. The reason for this is the "`expire`" option configured in each of the zone file's SOA section. So, what do you do when you *know* that a domain is going to be down for an extended period of time OR the domain has already been down for over a week and is now UNAVAILABLE?

  If you know AHEAD of time that the domain is going to go down:

  - Ask the administrator of the domain to edit the zone file(s) and increase the SOA's EXPIRE field to something LONGER than the time that the master DNS will be offline. This is configured in units of seconds. Don't forget to also tell the admin to update the zone file's SERIAL number and reload the DNS server to re-sync to all of the domain's secondary servers with these zone file changes.

  If the MASTER domain server is already down and there AREN'T any other master servers for this domain to make changes to the zone's SOA "`expire`" option, you only really have one option:

  - You need to change one of the secondary servers to now be a MASTER name server for that specific zone. To do this, simply change the name server's "`etc/named.conf`" file on the specific zone entry from "secondary" to "master". Also be sure you don't forget to allow zone transfers for this domain to that zone's other secondary name servers (as shown via whois).

24.20 Secondary DNS Design considerations

It should be mentioned that there is a very interesting and SERIOUS design issue that needs to be considered when setting up secondary zones with a split DNS setup. Say you have acme123.com running on both the INTERNAL -and- EXTERNAL processes on a server (same as the TrinityOS example set above).

The problem arises when you secondary for some remote domain(s) on the Internet. The email server for your domain then tries to send email to that remote email server. The process goes something as follows:

- Your internal SMTP server, which uses your INTERNAL DNS server (127.0.0.1) as its DNS server, does a DNS MX lookup for the destination email server.. say "buggs.com".
- So the internal DNS server (127.0.0.1) goes out to the Internet and asks, "what server is authoritative for the "buggs.com" domain". A response comes back saying: " your machine, ns.acme123.com is authoritative!" Technically, this is true. Well, HALF true actually.
- If you followed the TrinityOS example exactly, your EXTERNAL DNS server (ns.acme123.com) *IS* authoritative for both the "acme123.com" domain as well as "buggs.com" domain but the INTERNAL server is not. The INTERNAL server is only authoritative for the "acme123.com" domain (not "buggs.com")!
- What does that all mean? That means that when this MX DNS response comes back to the INTERNAL acme123.com server, the 127.0.0.1 server will think.. "Hey! They said I'm authoritative for that "buggs.com" domain but I don't know anything about it!" Error...
- If you had this situation, you would ultimately see weird and unhelpful error messages in the SYSLOG files that look something like: named[1188]: ns_forw: query(buggs.com) contains our address (roadrunner.acme123.com:192.168.0.1) learnt (A=acme123.com:NS=1.2.3.4)

Not very useful eh?

There are TWO valid solutions:

- **One:** You setup *both* the INTERNAL and EXTERNAL dns servers to secondary for the remote DNS zone(s). This would basically duplicate the secondary configurations from the EXTERNAL /home/chroot-dns-ext/etc/named.conf file into the INTERNAL /home/chroot-dns-int/etc/named.conf file. For example, you would copy these zone configurations from the EXTERNAL named.conf to your INTERNAL named.conf file. For example, you would copy info like the following into the Internal named.conf file:

```
zone "acme123.com." {
  type slave;
  file "acme123.com.db";
  masters { 100.200.0.212; };
  allow-transfer { none; };
};
```

  This would effectively make both the internal and external acme123.com DNS servers authoritative for those remote secondary zones. Now, when one of the slave DNS servers change something in their zones, both the external AND internal DNS processes would actually get a zone transfer.

- **Two:** You can change your internal zone name to something OTHER than being "acme123.com". Don't worry.. this won't hurt ANYTHING, even email as the Sendmail configuration shown in TrinityOS will re-write the email headers anyway. For example: you could change your internal domain from "acme123.com" to to "acme123.pvt". Yes, ".pvt". Remember, this is YOUR DNS server so, while only domains like ".com, .net, .org, .us, etc." are legal on the Internet (today), anything goes for internal networks. So, with this .pvt domain configuration in place, the internal DNS server would know that it is NOT authoritative for the "acme123.com" domain. Because it is no longer "acme123.com", it is also NOT authoritative for those other remote secondary zones ("buggs.com"). This might all seem like a pain but this second solution is somewhat cleaner than solution #1. Ultimately.. both work fine.

24.21 Automating the maintenance of the root-hints.db file

Ok, now DNS is hopefully working for your new connection. Next, I recommend that you implement the following script to maintain the root-hints file. Remember, the ROOT DNS server addresses change from time time. This script borrowed from the tldp.org's DNS-HOWTO (with a few changes on my behalf [should be in the DNS-HOWTO now]) makes sure things are occasionally updated:

/usr/local/sbin/root-hints-update

<root-hints-update START>

```
#!/bin/bash
#
# Part of the copyrighted and trademarked TrinityOS document.
# http://www.ecst.csuchico.edu/~dranch/LINUX/index-linux.html
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
# v2.6 - Fixed an error where the root.hints.new file was missing
#        from the "results" email.  The script is now deleting the
#        "results" file and is using all absolute paths.  Finally, the
#        script is again sending the "result" output as well.
# v2.5 - Fixed a filename error where the final status email was using
#        int/root.hints.new instead of int/root.hints.db
#      - Removed the line trying to delete a non-existant file
#      - Added some echo statements to make things a little
#        clearer
# v2.4 - Updated the dig info lookup from ns.internic.net to
#        a.root-servers.net
# v2.3 - Updated the initial CD into one of the real CHROOTed dirs
#        vs. /var/named.  The old script was also leaving a stray NEW
#        file in the EXT directory.  Because of all this, the email
#        notification would show an old root.hints file though DNS
#        would have the correct updated file.
# v2.2 - Change getting the hints file from rs.internic.net to ns.internic.
#        net
# v2.1 - Fixed a typo in the CHMOD of the external root-hints.sb file
#      - Fixed the file ownership of the internal root-hints.db file
#      - Changed the default path of where the new root.hints.new file
#        is to be placed
#      - Updated to have a backup copy of the INTERNAL hints file and not
#        just have an EXTERNAL backup
# v2.0 - Updated the script to support dual zone files
# v1.3 - Updated the script to show more verbose FAILURE logs.
#        Thanks to jon.marks@novatek.co.nz for the ideas
#
# v1.2 - added the test if no ROOT-SERVERS were returned
```

```
# v1.1 - added the test if the result had a SERV-FAIL
# v1.0 - original script from the DNS-HOWTO

echo -e "Running /home/chroot-dns/ext/var/named/root-hints-update..\n"
export PATH=/sbin:/usr/sbin:/bin:/usr/bin:

echo "Entering chroot-dns-ext"
cd /home/chroot-dns-ext/var/named

echo "Getting current root servers list.."
dig @a.root-servers.net . ns > /home/chroot-dns-ext/var/named/root.hints.new \
2> /home/chroot-dns-ext/var/named/result

DIG_OUTCOME=FAIL
if [ ‘grep -c SERVFAIL /home/chroot-dns-ext/var/named/root.hints.new ‘ = 0 \
] && [ ‘grep -c ROOT-SERVERS /home/chroot-dns-ext/var/named/root.hints.new‘ -gt 0 ]
   then
        DIG_OUTCOME=SUCCESS
        echo "   - Copying new hints file to the EXT named directory"
        mv -f /home/chroot-dns-ext/var/named/root.hints.db /home/chroot-dns-ext/var/named/root.hints.db.old
        cp -f /home/chroot-dns-ext/var/named/root.hints.new /home/chroot-dns-ext/var/named/root.hints.db
        chown chroot-dns-ext:chroot-dns-ext /home/chroot-dns-ext/var/named/root.hints.db
        chmod 444 /home/chroot-dns-ext/var/named/root.hints.db

        echo "   - Moving new hints file to the INT named directory"
        mv -f /home/chroot-dns-int/var/named/root.hints.db /home/chroot-dns-int/var/named/root.hints.db.old
        mv /home/chroot-dns-ext/var/named/root.hints.new /home/chroot-dns-int/var/named/root.hints.db
        chown chroot-dns-int:chroot-dns-int /home/chroot-dns-int/var/named/root.hints.db
        chmod 444 /home/chroot-dns-int/var/named/root.hints.db

        echo "Restarting both INT and EXT name.."
        echo -n "Restarting named: " >> result
        # note:  We dont use restart since old Redhat didn’t support it
        /etc/rc.d/init.d/named stop >> /home/chroot-dns-ext/var/named/result
        /etc/rc.d/init.d/named start >> /home/chroot-dns-ext/var/named/result
fi

echo "Emailing the results to root.."
(
        echo "To: hostmaster <root>"
        echo "From: system <root>"
        echo "Subject: TrinityOS DNS monthly root.hints.db update status: $DIG_OUTCOME."
        echo
        cat /home/chroot-dns-ext/var/named/result
        cat /home/chroot-dns-ext/var/named/root.hints.db
        echo

) | /usr/sbin/sendmail -t
echo "Done."
rm -f /home/chroot-dns-ext/var/named/result
```

```
exit 0
```

<root-hints-update STOP>

Now, make it executable and readable ONLY by the **root** user:

```
                chmod 700 /usr/local/sbin/root-hints-update
```

Finally, put it in the cron job to run monthly:

Redhat:

```
        ln -s /usr/local/sbin/root-hints-update /etc/cron.monthly/root-hints-update
```

Slackware:

- Edit "/var/spool/cron/crontab/root" and add this line to the bootom of the file:

```
                02 3 1 * *      /usr/local/sbin/root-hints-update
```

That's it!

24.22 How to acquire an Internet Domain Name

To get your own Internet domain, you need:

1. A pre-selected Internet domain name that isn't already taken. You can check to see if your desired domain is available by going to:

   `<http://www.internic.net>`

   or use the UNIX "whois" command.

   If the domain you want is already gone, don't forget to try the other suffixes like .com, .net, .org and now the new TLDs like .biz, .info, .name, .museum, .coop, .aero, and .pro. You should also know that many other countries are pushing users to use their domain space. For example, .cc and .tv are fairly popular with some people. **NOTE:** U.S. laws are about to change in the Internet. Currently, sleazy Internet users have been reserving domain names like cheezewiz.com and making the rightful owners (Kraft Corporation) pay ransoms to get them back.

   In 2000, companies that owned standard name trademarks to names, like CheeseWiz, finally got the LEGAL rights get to those domains. On the flip side, even if you had the domain superdupergizo.com for years and sold even gizmos with that name, someone might get that name "SuperDumoGizmo" trademarked. If that happened, they would then have the LEGAL right to take that domain away from you. Sucks huh?

   How can you protect YOUR domain? You might also want to get your domain trademarked. You might not care too much about this but some people will NEED TO. Please also understand that if you get a trademark for for the name and you already secured the .com domain name, you will then have legal grounds to kick people off the .net and .org domains as well. Personally, I think it will be cheaper in the long run if you just register ALL three domain name suffixes (.com, .net, .org) at one time. But if you then start to think about the new .biz, .info, etc. domains and this can be a LOT of money. Overall, the whole situation is a mess and I'm not sure what is the least-evil way of protecting your domain.

2. You need agreements with (1) or more EXISTING /remote/ DNS servers their to be your secondary (backup) DNS servers. You will have to coordinate this with the remote DNS administrators but it isn't too hard. It should also be noted that many Domain registrars can act a secondary DNS server for an addition fee. As it stands, the setup of the secondary DNS support is fully documented in TrinityOS's DNS section.

   - **NOTE:** You can RESERVE your desired DNS domain name NOW and not configure any DNS stuff for however long as you want. Basically, once you pay for for the domain, the domain is then YOURS unless you don't pay the renewal fees. One thing several Internet Domain Registrars are now doing is providing full co-location service for your domain where they will setup the DNS services, email, etc. ALL on their own servers for additional fees. Understand that these services costs more than just the purchase of the initial domain name procurment (currently $119 for 2 years from Network Solutions (Verisign) but some people like this service. Realistically, if you've read TrinityOS this far, you obviously want to run your own domain on your OWN server.
   - **NOTE #2:** Realistically, the primary and multiple secondary servers shouldn't be on the same network or preferably even through the same ISP. For example: if you want to put a DNS server behind your "XYZ" ISP provider, your secondary DNS servers shouldn't be connected via "XYZ" as well. Why? What happens if the ISP's network goes down? ALL DNS for your domain will fail. That means email will bounce, etc.

3. A permanent Internet connection with a static IP –OR– you can sign up with some of those dynamic DNS providers and THEY can then update their zones to you.

4. A credit card (makes things easier but they can also bill you too for bulk requests). Each domain currently costs different amounts depending which Registrar you use. DirectNIC charges $15 (U.S.) per domain but other Registrars might be even cheaper. Do your homework and see what you find. **NOTE:** Fortunately, you can usually deduct this cost from your taxes.

5. Now, with all this information (IP addresses, etc), go to `<http://www.internic.net>` and pick a Registrar. The incumbant registrar is Network Solutions (NSI) but my experience with them hasn't been very good. Though I can't recommend one registrar over another, I encourage you to research it a little. If you have good/bad luck with some of these new players, I'd love to hear from you.

6. Follow the prompts and enter in your domain name(s). Then click on either "reserve" or "register". **NOTE:** In the past, all DNS registrations were done via an email-only system. It was confusing at times and a pain. The new systems are usually SSL WWW based and is much easier to use. Interestingly enough, NSI would let you fill things out via a WWW form but it still would email you the completed form and then expect you to EMAIL it back to them. Lame. This might not be the case anymore as I don't use nor recommend NSI anymore.

   NOTE #2: Do not put in bogus data for any of the fields thinking it will keep your information private from SPAMMERs, etc.. Registrars check the info and if it doesn't all check, they will deny you the domain. They need your snail mailing address for your receipt and telephone numbers in case your DNS server, etc. goes down, is hacked into, etc. Them having your phone number is more valuable than you might think.

   NOTE #3: If you chose to use Network Solutions, you might be filling out the new Contact Information area, you might see the section for security. There are three types:

```
        MAIL-FROM:  This means that any changes to your domain must come
                              from an email address from your domain and it is
                              the default setting.

                              DO NOT USE THIS OPTION.
```

```
                                Its too simple for remote people to forge email.
                                Because of this, many people have had their domains
                                STOLEN from them because of this weak link.

        CRYPT:          This is a password encrypted setup.  This is pretty good
                                as long as you use a GOOD password.  See
                <ref id="sect-8" name="Section 8"> in TrinityOS for how to
                pick good passwords.

        PGP:            This is the ultimate in security and you need to submit your
                                public PGP key to the Internic.  BE WARNED:
                                If you change your PGP key often (your need to do this),
                                you might lock yourself out of your domain and you will
                                have to call the Internic direct.
```

If you DO NOT SEE these fields, don't worry. Once you finish your domain registration, go back to:

`<http://www.networksolutions.com/cgi-bin/itts/handle>`

and change it there.

7. When the Registrar asks you for an email address, do NOT use an email address that will be behind this new domain. Why? Until you get this DNS system fully running, any email from the Registrar sent to this email will be lost! Get it? Putting it another way, if you have problems with your domain and email isn't working, you WON'T be able to fix it because some registrars expect Domain change emails to come FROM the problem DNS domain. Stupid.. very stupid. Eh.. But.. don't worry, once everything works uine, you can go back and change this address.

8. After that, its pretty simple and VERY fast.

If you need more info on DNS, follow this great HOWTO:

`<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/DNS-HOWTO>`

25 SMTP MAIL: Sendmail configuration w/ domain masquerading & spam filters

Sendmail is one of the most common MTAs or Mail Transfer Agent email servers using on Linux. There are also several other viable email daemon alternatives like Postfix, Qmail, etc. So why did I initially pick and still STAY with Sendmail? Well, Sendmail is the most common email server out there and it's well documented. Some TrinityOS users also email me complaining that Sendmail is slow, bloated, or insecure when compared to other MTAs. In the past, this argument had some real truth to it but not with modern versions of Sendmail. Sendmail is now just as fast, secure, and probably MORE powerful than any other MTA out there. Ultimately, it's your decision but I think picking Sendmail is a good one.

Though configuring Sendmail and running might seem compilicated, it isn't too bad. Just take it a step at a time and you'll do fine. Yes, many of the commands are terse but the included configs are pretty good. If you don't trust TrinityOS's configs, check out `<http://www.sendmail.org>` for more details.

25.1 Determining what version of Sendmail you are running

```
********
**
**  Currently, Sendmail 8.12.9 and 8.11.7 (patched) are the latest known
**  SECURE versions of Sendmail though there is a KNOWN issue with the
**  "smrsh" shell.  This isn't an issue for the TrinityOS configuration but
**  patches are available if you need smrsh functionality.  If you are
**  running an older version, please UPGRADE.
**                                                     -------
**  If you aren't sure what version of Sendmail you are running or what
**  features were compiled into your version of Sendmail, try this command:
**
**      Generic method:      sendmail -d0.1 </dev/null
**
**  Redhat:             rpm -qa | grep sendmail
**
********

-----------------------------------------------------------------------------
NOTE:   The following Sendmail configs are:

        1. Tuned for Anti-SPAM via blackhole lists.  Please note that
           I'm 100% sure you will drop email from some of your friends
           because their ISP is associated with UCE or SPAM.  Until
           the SPAM situation improves, drastic measures like this are
           required.  It should be noted that I'm coming to the conclusion
           that these anti-spam blackhole systems DON'T work very well and
           cause more problems than they are worth.  Stay tuned as I'm
           not going to let this continue.

        2. Tailored to MASQ users that have 1+ machines on an internal LAN

        3. Users of Sendmail >= 8.9.x

           Sendmail 8.8.x users can find 8.8.x in the TrinityOS-Retired
           documentation available at:

<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS/RETIRED/TrinityOS-Retired.html>


        BUT these configs also apply to:

                 2. Linux users that are NOT doing MASQ will *STILL* need to make some
                        of the changes below if they plan to have their Linux box send
                        email whatsoever.
-----------------------------------------------------------------------------
```

25.2 Notes about changes in Sendmail over various versions of Sendmail

As Sendmail continues to evolve to fill the needs of various users, the configuration files, file locations, and mechanisms have changed. Here is a small table of the changes that effect TrinityOS users:

Sendmail 8.11.x+

- Local aliases = /etc/mail/aliases
- Local domain names = /etc/mail/local-host-names
- Backup SMTP domains = /etc/mail/access
- Correct Path and file permissions are required

Sendmail 8.9.x+

- Local aliases = /etc/mail/aliases
- Local domain names = /etc/mail/sendmail.cw

Sendmail 8.8.x

- Local aliases = /etc/aliases
- Local domains names = /etc/sendmail.cw

Distribution Specific

- Redhat - still puts sendmail.cf and aliases in /etc. You can either solve this via the sendmail.mc file or move the files into /etc/mail and symlink them back to /etc.

25.3 Downloading and either compiling or installing Sendmail from binaries

- Before you start installing a new copy of Sendmail already, backup your configs now:
  - Sendmail 8.11.x - 8.9.x+

```
        tar czvf /root/backup/sendmail-old.tgz /etc/aliases /etc/sendmail.* /etc/mail/* /usr/sbin/sendmail /usr/
```

Thoughts on the use of binary RPMs vs. compiling source code

- There are only two programs that I feel you you absolutely CAN NOT afford to screw up on: BIND (dns) and Sendmail (smtp)
- Because of this, install it hand (don't do binaries) and keep the configs current too. RPMs can't think for you and sometimes they mess up.

  With that said...

Installing via RPMs:

- Download the newest stable version of the Sendmail RPM code /and/ the associated Sendmail PGP signatures from the Sendmail URLs in 5 (Section 5). Put these files in, for example, the /usr/src/archive/sendmail directory.

- Verify that the PGP signature of the Sendmail source is ok (this step assumes you have GnuPG installed but not nessisarily be configured). `cd /usr/src/archive/sendmail`
  `gpg -import PGPKEYS`
  `gpg -verify sendmail.8.11.6.rpm`

  Make sure it says "Good Signature" at the top. There might be some trust warnings but don't worry about that.
- Next, I recommend to check out the RPM and see what it is going to install and/or possible OVERWRITE on your system. To do this, check out the top of 52 (Section 52)
- Now install the new RPMS:

```
rpm -Uvh sendmail-*.rpm
```

- Next, skip beyond the below compiling directions to properly configure Sendmail.

The recommended TrinityOS approach to **installing Sendmail is via COMPILING** it. See the "Thoughts" item in the RPMs paragraph above.

- Download the newest stable version of the Sendmail source /and/ the associated Sendmail PGP signatures from the Sendmail URLs in 5 (Section 5). Put these files in say, the /usr/src/archive/sendmail directory.
- Next, verify that the PGP signature of the Sendmail source is ok (this step assumes you have GnuPG installed but not nessisarily be configured). `cd /usr/src/archive/sendmail`
  `gunzip sendmail.8.11.6.tar.gz` `gpg -import PGPKEYS`
  `gpg -verify sendmail.8.11.6.tar.sig`

  Make sure it says "Good Signature" at the top. There might be some trust warnings but don't worry about that.
- Now uncompress the .tar file: `tar -xvf sendmail.8.11.6.tar`
- cd into the new sendmail's "src" directory
- Some rare users running older Linux distributions might need to edit the file "devtools/OS/Linux" and a line change it to

```
LIBS=   ifdef(‘confLIBS’, ‘confLIBS’)
```

  and change it to read:

```
LIBS=   ifdef(‘confLIBS’, ‘confLIBS’) -lresolv
```

  Save it.
- OPTIONAL - Though this step is optional, I recommend to HIDE the version of Sendmail you are running from the world. Though the `trinityos.mc` file shown below will hide this info from most Sendmail responses, it cannot do them all. THIS will and I bet it will help protect you from any current and even possible future Sendmail-specific Internet worms. Edit the file `sendmail/version.c` file and change the version number in the quotes to something like "TrinityOS Hardened".
- Now it's time to compile things up. Type in:

```
Sendmail 8.11.x+ : sh Build
       or
Sendmail 8.9.x   :  make
```

(If you have compiling problems, see `<http://www.sendmail.org/compiling.html>` for more info)

- Next, run the following to install Sendmail and all of its docs.

```
make install
```

If Sendmail is already running, shut it down :

- Redhat:

```
/etc/rc.d/init.d/sendmail stop
```

- Slackware:

```
kill -9 ‘ps aux | grep sendmail | grep -v -e grep | awk ’{print $2}’‘
```

Finally, I recommend to move over the new Sendmail docs to their proper resting place. For this example, I put Sendmail in /usr/src/archive/Sendmail/Sendmail-8.11.x and it will goto /usr/lib/sendmail-if/ :

```
cd /usr/src/archive/sendmail/sendmail-8.11.x/cf
tar cf - . | (cd /usr/lib/sendmail-cf/; tar xvf -)
```

25.4 Final install clean-up

Currently, Sendmail 8.12.9 and 8.11.7 have a "smrsh" security bug. It’s patchable but TrinityOS doesn’t use it. So, I recommend to just disable it by running:

```
chmod 500 /usr/sbin/smrsh
```

25.5 Configuring Sendmail to support your single or multiple Domain name(s)

Next, regardless if you are going to run a MASQ or non-MASQed network, edit or create the following. Please note that the /etc/mail/local-host-names is very important since it tells Sendmail WHAT DOMAINS TO ACCEPT EMAIL FOR. In this file, put in ****ALL**** of the domain names you registered with the Internic. Basically, /any/ hosts listed via the "whois" command for a given Internet domain you want to be the FINAL destination for should be listed in this file.

NOTE: If you are going to be a BACKUP email server (temporary email storage) for other domains, the hostnames of those remote servers for those domain names should NOT be listed in this file.

Sendmail 8.11.x - 8.10.x

```
/etc/mail/local-host-names
--
acme123.com
--
```

Sendmail - 8.9.x

```
                        /etc/mail/sendmail.cw
                        --
                        acme123.com
                        --
```

```
***********************************************************************
** Supporting more than one Internet domain - NOT being a backup MX
**
** If you are going to host MULTIPLE Internet domains on this one
** box (ie.  acme123.com and newdomain.com), simply add all
** the other domain names that you want to be able to receive
** email for in the files for your Sendmail version as shown above
** and you'll be set!
**
** This is NOT for being a backup email server for remote domains.
***********************************************************************
```

25.6 Configuring the Sendmail .mc files via m4 or by hand

```
        =================================================================
        All users, regardless of using the RPMs or compiling the source:
        =================================================================
```

- As of Sendmail 8.10.x, the various FILE and PATH permissions are now CHECKED. If the permissions aren't correct, Sendmail won't load. So, lets make sure they are correct. Run the following commands:

```
              chmod go-w / /etc /etc/mail /usr /var /var/spool /var/spool/mqueue
              chown root / /etc /etc/mail /usr /var /var/spool /var/spool/mqueue
```

- If you were to use Sendmail now, it would be **broken** since you would be sending mail from your machine but the receiver would see "ns.yourhost.com" in the reply field and NOT "yourhost.com". To fix this, you need to enable Sendmail's "domain masquerading" feature. You can do this the easy M4 way or the hard way (I recommend the easy way).

Doing it the M4 way (recommended):

- Sendmail's .cf example files and the .m4 scripting language need to be installed.

  RPM users: Verify that this package is installed by typing in "rpm -q sendmail-cf"

  Compiling users:

```
     mkdir /usr/lib/sendmail-cf
     tar cpf - /usr/src/archive/sendmail/sendmail-x.x.x/* | (cd /usr/lib/sendmail-cf; tar xpvf -)
```

- Go to /usr/lib/sendmail-cf/cf

  `Redhat users:  NOTE: You may or may NOT have this file Make a backup of your old .mc file cp redhat.mc redhat.mc.old`

- Create the "trinityos.mc" file.

  NOTE #1 - you only have to update the lines that have "acme123.com" in it. Leave the rest alone for LINUX systems.

All of the following configuration options are fully described in **/usr/lib/sendmail-cf/README**:

25.6.1 .mc Configs for Sendmail 8.11.x

/usr/lib/sendmail-cf/cf/trinityos.mc

```
--
#TrinityOS.mc 8.11.x config - v050402
#
#Give the configuration a version number
VERSIONID(`@(#)trinityos.mc       8.11 (Berkeley) 12/21/01')

#Tell sendmail that the CF file is for the Linux OS
OSTYPE(linux)

#Disable UUCP.  Its old and dead.
FEATURE(nouucp,reject)

#When sending email locally, use procmail to send mail vs. sendmail.  More efficient.
FEATURE(local_procmail)

#Enable the SMTP protocol - other options are the legacy protocols like UUCP and BitNet
MAILER(smtp)

#Use procmail as the local mailer.
MAILER(procmail)

#Rewrite ALL outgoing email to be from acme123.com and not somehost.acme123.com
MASQUERADE_AS(acme123.com)
MASQUERADE_DOMAIN(acme123.com)
FEATURE(masquerade_entire_domain)

#This also does the above trick but also works more in the header.
FEATURE(masquerade_envelope)

#If you email someone locally, say "greg" without the full domain, Sendmail will
#append acme123.com to the address.  "greg@acme123.com"
FEATURE(always_add_domain)


#Enable the use of the various Blackhole lists for automatic SPAM filtering
#
#  Make sure that each line is NOT wrapped.  Make sure its one long line
#
#  WARNING: This is tuned for Anti-SPAM via blackhole lists.  Please note that
#           I'm 100% sure you will drop email from some of your friends
```

```
#           because their ISP is associated with UCE or SPAM.  Until
#           the SPAM situation improves, drastic measures like this are
#           required
#
#  Note:    083003: Removed the use of relays.osirusoft.com since they are now gone
#
FEATURE(dnsbl, ‘bl.spamcop.net’, ‘Mail rejected - Open spam relay - see http://spamcop.net/bl.shtml? $&{client_ad
FEATURE(dnsbl, ‘unconfirmed.dsbl.org’, ‘Rejected - See http://unconfirmed.dsbl.org/’)dnl
FEATURE(dnsbl, ‘relays.ordb.org’, ‘Mail rejected - Open spam relay - see http://ordb.org/’)dnl


#Use the /etc/mail/sendmail.cw file for what domains to allow the receiving of
#email for.  This option is old and has been replaced with the /etc/mail/
#lost-host-names file
FEATURE(use_cw_file)

#Define where sendmail can find procmail
define(‘PROCMAIL_MAILER_PATH’, ‘/usr/bin/procmail’)

#Delete all the program and version information out of the SMTP header
define(‘confSMTP_LOGIN_MSG’,‘’)

#Enhance security by not offering version numbers in the HELP output
define(‘HELP_FILE’,‘’)

#Enable more secure operation of Sendmail
define(‘confPRIVACY_FLAGS’,‘authwarnings noexpn novrfy needmailhelo noetrn’)

#Enable the new Sendmail access DB support.. needed for backup SMTP setups
FEATURE(access_db)

#Enable to support backup SMTP for remote domains where the remote user is NOT locally defined
#on the local box
FEATURE(relay_mail_from)
--
```

25.6.2 Old .mc Configs for Sendmail 8.9.x

```
*****************************************************
* Please do NOT use old versions of Sendmail unless  *
* ABSOLUTELY required to void spam and possible      *
* security issues!!                                  *
*****************************************************
```

/usr/lib/sendmail-cf/cf/trinityos.mc

```
--
#TrinityOS.mc 8.9.x config - OBSOLETE - do NOT use
#
#Give the configuration a version number
```

```
VERSIONID(`@(#)trinityos.mc       8.10 (Berkeley) 11/26/99')

#Tell sendmail that the CF file is for the Linux OS
OSTYPE(linux)

#Disable UUCP.  Its old and dead.
FEATURE(nouucp)

#When sending email locally, use procmail to send mail vs. sendmail.  More efficient.
FEATURE(local_procmail)

#Use procmail as the local mailer.
MAILER(procmail)

#Enable the SMTP protocol - other options are the legacy protocols like UUCP and BitNet
MAILER(smtp)

#Rewrite ALL outgoing email to be from acme123.com and not somehost.acme123.com
MASQUERADE_AS(acme123.com)
MASQUERADE_DOMAIN(acme123.com)
FEATURE(masquerade_entire_domain)

#This also does the above trick but also works more in the header.
FEATURE(masquerade_envelope)

#If you email someone locally, say "greg" without the full domain, Sendmail will
#append acme123.com to the address.  "greg@acme123.com"
FEATURE(always_add_domain)

#Enable the use of the Realtime Blackhole list for automatic SPAM filtering
FEATURE(rbl)

#Use the /etc/sendmail.cw file for what domains to allow the receiving of
#email for.  This option is old and will be replace with something else.
FEATURE(use_cw_file)

#Define where sendmail can find procmail
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')

#Delete all the program and version information out of the SMTP header
define(`confSMTP_LOGIN_MSG',`')

#Enable more secure operation of Sendmail
define(`confPRIVACY_FLAGS',`authwarnings noexpn novrfy needmailhelo noetrn')
--
```

The following script will create the "trinityos.cf" file from the just created "trinityos.mc" file. I recommend you save this script so you don't have to type all this in every time you change something in the .mc file.

/usr/lib/sendmail-cf/cf/generate-cf

```
#!/bin/sh
```

```
# TrinityOS - generate.cf script - v050402
#
CFDIR="/usr/lib/sendmail-cf"
SRCFILE="trinityos"

cd $CFDIR
m4 ${CFDIR}/m4/cf.m4 ${CFDIR}/cf/$SRCFILE.mc > ${CFDIR}/cf/$SRCFILE.cf

# Please note this is the destination directory for Sendmail 8.9.x and
# newer
if [ -f ${CFDIR}/cf/$SRCFILE.cf ]; then
  mv /etc/mail/sendmail.cf /etc/mail/sendmail-`date +%m%d%y`
  cp ${CFDIR}/cf/$SRCFILE.cf /etc/mail/sendmail.cf
  echo -e "New CF file created.\n\n `ls -la /etc/mail/sendmail.cf`\n"
  echo -e "Restart Sendmail for changes to take effect\n"
 else
  echo -e "\nError: Output CF file not found\n"
fi
```

Doing it the hacker way (NOT recommended unless you really REALLY know what you are doing:

- \- Manually edit the /etc/mail/sendmail.cf
- \- Near line 164, you will see "DM" by itself. Add your domain to this line. e.g.

```
DMacme123.com
```

- \- Near lines 813 and 814, change the terse lines from Sendmail section S94:

```
S94
#R$+                    $@ $>93 $1
R$* < @ *LOCAL* > $* $: $1 < @ $j . > $2
```

to this:

```
S94
R$+                    $@ $>93 $1
#R$* < @ *LOCAL* > $*  $: $1 < @ $j . > $2
```

25.7 Email Alias and Relay configuration

In the future, 18 (Section 18) of TrinityOS will be inserted here. Until then, please jump to that section to make sure you have any required email aliases setup.

25.8 Configuring DNS MX records

The final step to setting up a email server is DNS. Basically, when you send an email to say "root@acme123.com", the sender's email program has to know what IP address to send this email to.

What happens is the sender's email program will first go out to the Internet and get an IP address of a DNS server that can answer for the "acme123.com" domain. Once this IP address is found, the email program will then ask for an "MX" record for this domain. An MX record or "Mail eXchange" host is basically a record of what hosts will accept email for this domain. You can have as many MX records in DNS as you want. Just be sure the hosts listed are setup to accept email for your domain. In addition to the host name for the MX record, there is a METRIC with each MX record. Lower the MX metric, the more the remote email server will be preferred over the other email servers. Basically, your machine should have the lowest MX metric and all of your backup email servers should have a higher metric.

Anyway, please see 24 (Section 24 - DNS) for all the specifics on configuring the DNS MX records. Please take SPECIAL note of secondary DNS servers section. If your DNS zone becomes unavailable due to your DNS server being down too long, it won't matter if you have several redundanct email servers or not. If the remote email clients can't resolve the MX record, the mail will bounce.

25.9 Some Possible Sendmail Startup Troubleshooting

1) Did you follow the "aliases" instructions in 18 (Section 18)?

2) Enable Debugging:

Sometimes you will need to run Sendmail in debugging mode to see what is really going on. To do this, follow these steps:

- Stop Sendmail:
  - Redhat: /etc/rc.d/init.d/sendmail stop
  - Slackware: kill -HUP ‘ps aux | grep sendmail | grep -v -e grep | awk ’{print $2}’‘
- Start Sendmail in forground debugging mode
  - /usr/sbin/sendmail -bD -d 30
    - Option "-bD" will make Sendmail load only in the foreground and -d by itself only enables debugging on a level of "9". Setting it to "30" is more helpful.
- When done with debugging, simple hit Control-C to stop Sendmail.
- Don't forget to restart Sendmail in daemon mode:
  - Redhat: /etc/rc.d/init.d/sendmail start
  - Slackware: /usr/sbin/sendmail -bd -q1h &

3) I had some issues with the 8.9.3 installation at this point. Specifically, I was getting the following in /var/log/maillog:

```
Aug 24 22:38:45 trinity2 sendmail[7375]: WAA07051: SYSERR(root): Cannot exec /usr/local/bin/procmail: No such fil
Aug 24 22:38:45 trinity2 sendmail[7368]: WAA07051: to=<dranch at trinnet dot net>, delay=00:10:10, xdelay=00:00:0
  stat=Operating system error
```

This is because sendmail wasn't looking for procmail in the right place. You can either implement the following hack or fix it the proper way by using the:

```
define(‘PROCMAIL_MAILER_PATH’, ‘/usr/bin/procmail’)
```

paramter in the 8.9.x. trinityos.mc file and then recompile the M4 script into a new resulting sendmail.cf file as shown earlier in this section.

To hack it and just get things running, I had to fix a path ISSUE:

```
        ln -s /usr/bin/procmail /usr/local/bin/procmail
```

25.10 Tuning Sendmail for security

Ok, next, you need to make sure that your mail server is SECURE and RELAY-free:

- When hackers want to hack into a given a email server, they will first want to find out what version of the email server you are running. Once they know what version you are running, they can then run exploits against it. Also, they will try to find out where root and postmaster email goes to. So, what can you do?

1. Always run the newest version of your email server. Be it Sendmail, Qmail, PostFix, etc.

2. Hide the name and version of your email server:

- Sendmail:

- Best method: The trinityos.mc file already uses the "define('HELP_FILE')" method to block remote users from MOSTLY determining what version of Sendmail you are running.
- Manual Method: The manual way requires you to edit the /etc/sendmail.cf file and change the following lines from:

```
            O SmtpGreetingMessage=$j Sendmail $v/$Z; $b

            O Privacy Options=authwarnings

            O HelpFile=/usr/lib/sendmail.hf
```

to:

```
            O SmtpGreetingMessage=

            O Privacy Options=authwarnings noexpn novrfy needmailhelo noetrn

            O HelpFile=
```

NOTE: The "Privacy Options" and "HelpFile" changes were already done for you in the above `/usr/lib/sendmail-cf/cf/trinityos.mc` file.

A note on Compatiblity :

I have had one user that told me that the "needmailhelo" option was possibly causing "SMTP error 250 - remote protocol error" problems with some remote SMTP servers. Please understand that this is NOT a Sendmail problem on your end. This option exposed a broken SMTP on the remote end.

You should also keep in mind that Sendmail, to this day, is one of the most tolerant SMTP servers when communicating to broken remote SMTP servers. If you were to move over to a different SMTP server, say Qmail, you would notice a LOT more broken SMTP servers out on the Internet.

25.11 Running Sendmail as a daemon or as a cron job

- Do you need Sendmail to run as a DAEMON:

You now need to determine if you need to have sendmail running all the time or just have it occasionally load up to send email. What's the difference?

- Sendmail ONLY needs to be always running if you have your own FQDN domain such as acme123.com which you registered with the Internic.

If you do have your own domain and want to receive email, make sure to enable Sendmail that was DISABLED in 8 (Section 8)

If you DON'T have your own domain, you DO NOT NEED Sendmail to always run. Because of this, I recommend to disable Sendmail as a DAEMON as shown in 8 (Section 8). If you do disable Sendmail but if you want to SEND email from your Linux box, you still need to have Sendmail (or any other MTA like Qmail, Vmail, PostFix, etc) installed.

If you aren't going to have Sendmail running Daemon mode, your locally sent email should be able to get out fine. But, if there is a problem with your Internet connection, the Internet itself, or the remote mail server, when you originally tried to send that mail, it WON'T be automatically be re-scheduled to be sent at a later time. To get Sendmail to retry later, you need to configure "cron" to try to resend any queued email once an hour.

To have sendmail try sending delayed email:

Redhat:

Create the /etc/cron.hourly/sendmail file

```
/usr/sbin/sendmail -q
```

Slackware:

edit the /var/spool/cron/crontabs/root file and add a line:

```
01 * * * * /usr/sbin/sendmail -q
```

Now, re-load cron to see the changes:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}’‘

25.12 Testing your Sendmail setup

That's it! Now you need to test Sendmail:

```
1. First, start it up:

        Redhat: /etc/rc.d/init.d/sendmail restart

        Slackware:      /usr/sbin/sendmail -bd -q1h
```

```
    2. If you are running your own domain:

            2.A. Send an email to the "root" account of your domain (for
     example: root@acme123.com) from a remote computer out
     on the Internet somewhere.  Make sure that this test email
     arrives to /your/ INBOX and not root's mailbox.

            2.B. Look at the email headers and make sure that the TO: field
     looks ok.


    3. Regardless, if you DO or DON'T have your own Internet domain name:

            3.A.  Send email /from/ the local Linux box to a different user on
                    the local Linux box (via Pine, ELM, etc).  Make sure it gets
                    there.

            3.B.  Send email from the local Linux box to the "root" account.
                    Make sure that this email is properly forwarded to the user
                    configured to receive "root's" email via
    Section 18


    4. For users that send email via a POP3/IMAP client (Eudora, Netscape,
etc) from an INTERNAL MASQed LAN connection:

            4.A. Be sure to configure your POP3/IMAP client properly.

            4.B. Send an email to a remote email account that you have
                    access to or that someone can then forward BACK to you.

            4.C. -LOOK- at the email headers.  Some programs make you
                     push some button to look at this information.  Eudora needs
     the "BlahBlah" button pushed.  Pine requires that
     you hit "O" for Options and then "H"
     for Header Mode (note: these PINE options must be ENABLED in
     Pine's configuration menus to even see them).

            4.D. Make sure that none of the To, From: Reply, etc. addresses
     look odd.


    5. For users that send email from a POP3/IMAP client (Eudora, Netscape,
       etc.) via the Internet (you are dialed into some other ISP, etc)

            5.A. Be sure to configure your POP3/IMAP client and Linux POP/IMAP
     server properly.

            5.B. Be sure that you can receive email via POP/IMAP from your
     Linux server.
```

***   5.C. Send a piece of email to a remote account via the local mail tools like Pine, elm, etc. Can you do it?   Probably not!!

The reason for this is because you are trying to to EMAIL RELAY through your Linux server and this is BAD. This is how you get a majority of all that SPAM email.

To fix this, add ANY remote network names, either INTERNAL or EXTERNAL, that you want to send email FROM into the /etc/mail/relay-domains file. For example, say I'm dialed into an ISP, say earthlink.net, and I want to send email via my Linux server. Also, I will want to send email from ANY machine on the internal MASQ'ed network. For this to work, I would have to do the following:

```
--/etc/mail/relay-domains
earthlink.net
192.168.0
--
```

This can also be done by adding the specific hosts or IPs to the /etc/mail/access file and marking them as "RELAY"s.

NOTE #1: I hope you realize that by doing line #1, any OTHER users that use Earthlink.net can ALSO use your Linux server as a relay site. This is BAD but you might not have any choice. Your only other (but preferred) choice is to get a STATIC IP address from your ISP (ie. Earthlink) and then configure in THAT specific name or TCP/IP address.

NOTE #2: For the second line, you can also add either the generic network IP address, a specific internal machine's IP address, your top level FQDN, (acme123.com), or the FQDN of each internal machine. Your pick.

6. Verify that the Blackhole Anti-Spam filter system is working. Run the following command from the command line:

```
--
$ sendmail -bt -C /etc/mail/sendmail.cf
  ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
  Enter <ruleset> <address>

> .D{client_addr}127.0.0.1
> Basic_check_relay <>

Basic_check_rela   input: < >
Basic_check_rela returns: OKSOFAR
```

```
> .D{client_addr}127.0.0.2
> Basic_check_relay <>

Basic_check_rela   input: < >
Basic_check_rela returns: $# error $@ 5 . 7 . 1 $: "550 Mail from " 127 . 0 . 0
. 2 " refused by blackhole site rbl.maps.vix.com"

> CTRL/D
--
```

Ahhh.. works like a charm!

7. Make sure that the online HELP system doesn't work:

   7.A TELNET to either your external IP, localhost, or internal IP address (if you have one) on port 25 and issue the HELP command. Type in QUIT when finished.

   ```
   telnet localhost 25
   --
   Trying 127.0.0.1...
   Connected to localhost.
   Escape character is '^]'.
   220  ESMTP

   HELP

   502 5.3.0 Sendmail TrinityOS -- HELP not implemented
   quit
   221 2.0.0 roadrunner.acme123.com closing connection
   Connection closed by foreign host.
   --
   ```

   7.B You will probably notice that the Sendmail version will show up when you do that "HELP" test. Please note that deleting all references to the Sendmail version numbers is difficult but not impossible if you have a minimal or decent understanding of C code. If you want to delete this specific instance, edit the Sendmail srcrsmtp.c file and search for "502 5.3.0". There, delete the "%s" from that line. You can replace it with anything you wish. As you can see above, I put in "TrinityOS". :)

8. Send a peice of email the manual way:

   8.A TELNET to your EXTERNAL IP address on port 25. From here, send email from some known good email address to yourself on your new email server.

```
telnet 102.200.0.25 25
--
Trying 102.200.0.25
Connected to roadrunner.acme123.com
Escape character is '^]'.
220  ESMTP

helo dranch
250 ns.acme123.com Hello roadrunner.acme123.com [100.200.0.212], pleased to meet you

MAIL FROM: <dranch@backupacme.com>
250 2.1.0 <dranch@backupacme.com>... Sender ok

RCPT TO: <dranch@acme123.com>
250 2.1.5 <dranch@acme123.com>... Recipient ok

data

354 Enter mail, end with "." on a line by itself
SUBJECT: email test

This is a manual TELNET test of email.

.
250 2.0.0 fBUH8t219012 Message accepted for delivery
quit
221 2.0.0 roadrunner.acme123.com closing connection
Connection closed by foreign host.
--
```

25.13 More troubleshooting help

Errors in the logs:

- If you get an error in the logs that says:

```
mail loops back to me (MX problem?)
```

This means that the machine doesn't know that HOST or DOMAIN. You might have a slightly different configuration than described in TrinityOS. To fix this, make sure you have EVERY permutation of the Linux server's DOMAIN and HOSTNAME in the /etc/mail/local-host-names. For example:

```
acme123.com
ns.acme123.com
roadrunner.acme123.com
```

Once you have changed this, restart Sendmail and try again.

25.14 Being a Backup SMTP email server (Backup MX) for other Internet domains

Why be a backup SMTP server? Well, if your email server or someone else's email server goes down (Internet connection breaks, power loss, etc.), a backup server will queue up your emails until the original email server is back up. There are several other possible reasons:

- So say YOU or a friend is changing ISPs and he/she needs another SMTP email server to queue email for his/her domain(s) while they are transitioning ISPs, IP addresses, updating the InterNIC, etc. as described in 52 (Section 52).
- You or a friend running an email server had his/her HD crash. With a backup email server, they can take their time getting things running again without losing any email.

Regardless of the reason, here are the steps to configure your Sendmail SMTP server to accept email for other domains. Please note that DNS changes and some backup DNS server is REQUIRED to get this running. Those changes are highlighted in 52 (Section 52)- "Gracefully transitioning Internet domains through a IP address or ISP change change".

Before we get started, you should understand a little terminology:

- Per the SMTP RFCs, an email will only be valid for FIVE days. So, even if you have a backup email server running for a given a domain, if the email is not delivered to a /final/ destination within five days, the email will be bounced (and returned to the original sender). The only solutions for this problem is to (1) setup a SMTP server to temporarily store email for this domain (common RELAY setup), (2) setup a SMTP server to ACCEPT the email on a temporary basis (become the autoritative email server for the domain), or (3) re-write the dates in the various emails so they won't expire. Overall, the first method is the normal situation and is recommended to be setup for EVERY domain. You never want to lose email. The second option is a realistic way to accept the remote email an then forward it to somewhere else until their remote email server is back online. Finally, option three is fairly radical and isn't recommended.
- RELAY: When you RELAY email for some domain, the backup server will temporarily store those emails. Every hour, the backup SMTP server will try to re-deliver those emails to the final destination for up to FIVE days. After five days, those emails will be "bounced" back to the original server telling them that the mail could not be delivered.
- FINAL DESTINATION: Unlike being a RELAY, being a FINAL DESTINATION for an Internet domain is no different than addition an additional domain to your own server. The difference is that you will use the `/etc/mail/aliases` take these emails and forward them to some OTHER email address.

  NOTE: It's important to NOT have have ANY of the remote domain(s) you are trying to be a final destination for be listed in the `/etc/mail/local-host-names`. If they are, your email server won't accept them as a final destination but try to relay them back to the down server. Understand?

To allow Sendmail to RELAY email for a different domain than your own, you first need to be sure that you enabled the `"FEATURE(access_db)"` and `FEATURE(relay_mail_from)` options in the trinityos.mc Sendmail M4 script shown earlier in this section. Once you are sure those options are present, compiled into the resulting `/etc/mail/sendmail.cf` file, follow these steps:

- The first step it to edit the **/etc/mail/access** file and add any remote domains you wish to be a SMTP RELAY/BACKUP for. The following example shows your server will be a BACKUP MTA for two remote domains:

```
# by default we allow relaying from localhost...
localhost.localdomain          RELAY
localhost                      RELAY
127.0.0.1                      RELAY

some-remote-domain.com         RELAY
yet-another-domain.net         RELAY
```

- Once this is configured, you need to compile up a new ACCESS datebase. Do this by running:

```
makemap hash /etc/mail/access < /etc/mail/access
```

I would also recommend to make this above line into a simple script (call it `/etc/mail/make-new-access` for future use).

That's it. Everything SHOULD work ok but you NEED to test it. To test it, follow the steps in Section 25.9.8 above but instead of TELNETing to the 127.0.0.1 address, TELNET to your one of your external backup MX email servers. If the server accepts your email and if you ultimately get the email on your own email server, then things are working FINE.

26 NTP Time calibration

Some of you might be wondering why didn't originally use to support XNTP. Why? Getdate is 37k with ALL the sources and compiled binaries where as Ntp-4.0.72i is over 8.8MB! For fricken just time calibration! Yes, Xntp does a LOT more than getdate but for the purposes we need here, it is MASSIVE overkill. But, many distributions come with it built-in so I will support it now.

I've been also told that newer versions of Slackware comes with "netdate" which is supposed to be just as good as "getdate". Since this only exists on Slackware, I'll stick with getdate and xntp for now.

IMPORTANT:

- It is good etiquette to email the NTP clock manager and confirm that its ok to sync off their clock server. These servers get POUNDed and many NTP managers will ban you from syncing to them unless you ask. Don't ask me why they get so uptight but some just do.

Redhat Users:

- If you time is WAY off regardless of using NTP or not, make sure the settings in /etc/sysconfig/ntp are correct.

- Download "xntpd" or "getdate" (URLs in 5 (Section 5) and put it in /usr/src/archive

Compiling Getdate:

- Uncompressit via "tar -xzvf
- Edit the Makefile
- Change the "PREFIX" to be /usr/local

- Run "make", "make install", "make installman"

Compiling Xntp:

- The compiling of Xntp has not been completed yet though most distros come with it pre-installed

  Now, go to 5 (Section 5) and pick a NTP server closest you. Test that it is up by running "getdate your.ntp.site". For example:

```
                    getdate ntp.nasa.gov
```

You should see output similar to:

```
                    ntp.nasa.gov:   (-68)   Sun Jun 14 10:27:28 1998
```

26.1 - The Getdate way:

- Edit the /usr/local/sbin/getdate file and make it look like so:

For example, this is what I use. Edit it to use servers local to you

/usr/local/sbin/get-date

```
#!/bin/sh
#
# Version: 07/03/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates:
#
# 07/03/00 - Added comments for users who want to save the date in UTC
#
# The "clock" command sets the CMOS clock time as well.
#
timehosts="otc2.psu.edu wwvb.erg.sri.com ntp.nasa.gov"
#

if /usr/local/bin/getdate -adjust 10 200 $timehosts > /dev/null; then
        /sbin/clock --systohc

    # NOTE: If you want to set your local to UTC, append "--utc" to the
    #       above "hwclock" line
fi
```

26.2 - The xntp way:

- Edit the /usr/local/sbin/set-clock file and make it look like so:

For example, this is what I use. Edit to use servers local to you

/usr/local/sbin/set-clock

```
--
#!/bin/sh
#
# Version: 07/03/00
#
# Part of the copyrighted and trademarked TrinityOS document.
# <url url="http://www.ecst.csuchico.edu/dranch">
#
# Written and Maintained by David A. Ranch
# dranch at trinnet dot net
#
# Updates:
#
# 07/03/00 - Added comments for users who want to save the date in UTC
#
# The "clock" command sets the CMOS clock time as well.
#
timehosts="otc2.psu.edu wwvb.erg.sri.com ntp.nasa.gov"
#
if /usr/sbin/ntpdate -ub $timehosts > /dev/null; then
    /sbin/hwclock  --systohc

    # NOTE: If you want to set your local to UTC, append "--utc" to the
    #       above "hwclock" line
fi
--
```

There are TWO examples shown here:

- NTP to run ONCE an hour
- NTP to run EVERY 15 minutes.

I recommend the once-an-hour method. The 15 minute method is primarily for users running Diald since the NTP traffic will bring up the link every 15 minutes.

- Slackware users:

- Edit "/var/spool/cron/crontab/root" and add this line to the bottom of the file:

- 60 minutes with "xntp"

```
* 0-23 * * *      /usr/local/sbin/set-clock
```

- 60 minutes with "getdate"

```
* 0-23 * * *      /usr/local/sbin/get-date
```

- 15 minutes with "xntp"

```
0,15,30,45 * * * *      /usr/local/sbin/set-clock
```

- 60 minutes with "getdate"

```
0,15,30,45 * * * *      /usr/local/sbin/get-date
```

- Lastly, tell CRON to re-read it's configuration file by running:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk '{print $2}'‘

- Redhat users

- 15 minutes

- Edit the /etc/crontab file and ADD this line ABOVE the cron.hourly line.

```
0,15,30,45 * * * * root run-parts /etc/cron.15min
```

- Link the script

```
ln -s /usr/local/sbin/get-date /etc/cron.hourly/get-date
```

- Tell CRON to re-read it's configuration file by running:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk '{print $2}'‘

- 60 minutes

- This hourly cron directory is already setup in Redhat

- Link the script

- 60 minutes the "xntp" way

```
ln -s /usr/local/sbin/get-date /etc/cron.hourly/set-clock
```

- 60 minutes the "getdate" way

```
ln -s /usr/local/sbin/get-date /etc/cron.hourly/get-date
```

- 15 minutes the "xntp" way

```
ln -s /usr/local/sbin/get-date /etc/cron.15min/set-clock
```

- 15 minutes the "getdate" way

```
ln -s /usr/local/sbin/get-date /etc/cron.15min/get-date
```

27 DHCPd SERVER configuration

DHCP is an automatic IP addressing tool much like BOOTP is. With DHCP, IP addresses don't have to be statically addressed and possibly manually changed on EACH computer in the future. DHCP can simply give out IP addresses but also configure many other options as well (see below). It's really a powerful mechanism. For more DHCP info including other URLs,etc., check out the DHCP section in 5 (Section 5).

Critical Note:

- You need to make sure that you are running DHCPd version 3.0p1 or newer as there are server local root exploits due to the newer Dynamic DNS update code. Please see your Linux Distribution's FTP server for DHCPd updates or 5 (Section 5) for URLs on downloading newer versions.

27.1 The Differences between DHCP and BOOTP

DHCP or Dynamic Host Control Protocol is the direct cousin of BOOTP.

- BOOTP: Bootp is usually used to give network equipment an IP address (usually static) and it also is used to initiate TFTP (trival file transfer protocol) file transfers to give this network equipment its operating system and possibly its configuration as well.
- DHCP: This newer protocol is more intended for computers on a given LAN for things like:

```
                                - Host name and FQDN
                                - IP address, mask and default gateway
                                - DNS servers
                                - WINS servers (optional)
                - NTP time servers
                                - etc.

The Internet peoples at be realized the BOOTP protocol was fairly inflexible
and wouldn't grow with new features.  So DHCP was created to be a flexible
protocol that, much like PPP, has negotiated parameters.  It can send out
everything from IP addresses to NTP servers.  DHCP is a great system to be
able to just plug a DHCP-compatible computer and DHCP will configure its
whole network system ON THE FLY.

DHCP is very flexible.  You can give it pools of dynamic IPs to give out,
statically give certain machines STATIC IPs (like below), etc.
```

  For more information, please see the DHCP RFCs in 5 (Section 5).

27.2 Configuring DHCP support on various Linux Distributions:

Though TrinityOS primarily supports Redhat, I'm contantly adding support for other Linux distributions. If you have additions or comments, please let me know.

- Redhat: - Make sure that the /etc/rc.d/rc3.d/S65dhcpd exists If not, enable it as defined in 8 (Section 8)

  - [OPTIONAL] - Edit the file /etc/rc.d/init.d/dhcpd and change the following.

NOTE: The following configuration is a requirement for 2.0 and 2.2.x kernels. It shouldn't be required for 2.4 and 2.6 kernels.

NOTE2: This configuration assumes you want to serve DHCP leases ONLY on the "eth1" interface

Start section line from:

– daemon dhcpd – to – route add -host 255.255.255.255 eth1 daemon dhcpd eth1 –

NOTE: You need to change the "interface" name to whatever INTERNAL LAN interface you want DHCP to run on. i.e. You DON'T want DHCP to run on your Internet connection!!

- Slackware: - [OPTIONAL] - Edit the file /etc/rc.d/init.d/dhcpd and change the following.

  NOTE: The following configuration is a requirement for 2.0 and 2.2.x kernels. It shouldn't be required for 2.4 and 2.6 kernels.

  Add the following line to the /etc/rc.d/rc.inet1 file:

  route add -host 255.255.255.255 eth1

  Add a line to execute dhcpd in the /etc/rc.d/rc.local file like:

  /usr/sbin/dhcpd eth1

- TurboLinux: TurboLinux uses ISC's /sbin/dhclient instead of the traditionaly used Linux clients.

  The configuration file for dhclient is found in **/etc/dhclient.conf** and control shell script of **/etc/dhclient-script**. This script has provisions to source a user-defined /etc/dhclient-exit-hooks file which it executes if found. Putting it simply, you can simply add the line "/etc/rc.d/init.d/firewall restart" to the /etc/dhclient-script file to properly load the firewall upon various DHCP events.

27.3 Determining MAC addresses for static DHCP scopes

NOTE: This config defines a STATIC IP address per core machine. All other machines get dynamic DHCP IP addresses. I do this for security reasons.

To find out the MAC address of a machine's Ethernet card, do the following:

Win95: run "winipcfg" WinNT: run "ipconfig /all" Linux: run "arp"

- For ALL distributions using the DHCPcd client, create and modify the file /etc/dhcpd.conf

27.4 Creating the /etc/dhcpd/conf file

```
--<begin>--
server-identifier roadrunner-int.acme123.com;

#Default ISC lease file path is /var/state/dhcp but Redhat is /var/dhcpd/
lease-file-name "/var/dhcpd/dhcpd.leases";
default-lease-time 86400;

#Disable all Dynamic DNS functionality
ddns-update-style none;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.1, 24.1.64.33, 24.1.64.34;
option domain-name "acme123.com";
```

```
subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.9 192.168.0.10;
}

host coyote.acme123.com {
        hardware ethernet 00:60:08:B1:36:4A;
        fixed-address 192.168.0.4;
}
--<end>--
```

Next, you need to create the dhcp.leases file:

```
                "touch /var/dhcpd/dhcpd.leases"
```

As mentioned above, you will need to replace the hardware Ethernet MAC addresses with the MAC addresses of your specific NIC cards.

* Ok, now you need to put in all of your DHCP IP addresses into DNS as described in 24 (Section 24) and then restart Bind.

Now, you need to make sure you have the following lines in your /etc/services file:

```
                --
                bootps          67/udp                          # bootp server
                bootpc          68/udp                          # bootp client
                --
```

27.5 Starting up DHCP

Finally, lets start DHCP up:

Slackware: Run "/usr/sbin/dhcpd eth1"

Redhat: Run "/etc/rc.d/init.d/dhcpd start"

* Additional security: DHCPd runs as root in a non-chroot'ed way. If you are paranoid about security, check out the LASG doc. The URL is in 5 (Section 5)

If that works well, you should enable DHCP full time:

Redhat:

```
                        chkconfig --level 2345 dhcpd on
```

27.6 Using DHCP Relay for LANS seperated by routers

Ok, so say that you have a network that you'd like to enable DHCP on but it is seperated by a router. Without any specical configuration, the DHCP client would send DHCP requests to the BROADCAST network address (255.255.255.255). The problem is that routers, by definition, surpress network broadcasts (all ones or 255.255.255.255). How do you solve this? Most modern routers support a feature called "DHCP Relay" (Juniper calls it "dhcp-relay" and Cisco calls it it "ip-helper address") which is a form of a DHCP proxy server. To read up on this, check out RFC 1542 in 5 (Section 5).

What a DHCP Relay agent does is record the originating network address of the requesting DHCP client and re-sends it out on the segment where the DHCP server is. In addition to this, the router with embed the IP address of the router's local IP address in GIADDR field of the DHCP packet.

When the DHCP server figures out what IP address to give to the remote DHCP client, it sends it back to the IP as created in the above GIADDR field. The router will receive this DHCP reply packet where the router will then re-transmit the DHCP reply on the original requesting DHCP network. Voila!

So how do you configure the Linux DHCP server to work with DHCP Relay enabled network(s)? You basically configure NOTHING! Huh? How does that work? When the DHCP server receives a DHCP request, it looks at the SRC IP address and the GIADDR field within the packet. If that SRC IP network MATCHES a configured "subnet" DHCP scope as configured in the dhcpd.conf file, it simply gives an IP address from that particular scope vs . a different one found elsewhere in the dhcpd.conf file. The one thing to note is that if the DHCP server is on the same network that it will be also serving DHCPed IP addresses to, just make sure that local "subnet' confuration stanza comes FIRST in /etc/dhcpd.conf file.

28 POP3 and IMAP4 e-mail services

First, a quick description of the various email client protocols:

UUCP: UUCP or UNIX-to-UNIX-COPY is the oldest email system out there and I doubt many use people anymore. Before the days of SMTP, it was the only game in town and VERY complicated.

POP3: POP3 or Post Office Protocol 3 is the older method get email but its still in use today. The issue with POP3 mail is that users authenticate to it in CLEAR TEXT. This is a bad thing. Fortunately, there are security add-ons to encrypt this username/password such as APOP, MD5, and even Kerberos.

Anothering thing to be aware about POP3 email is that the client will actually download ALL the email from the server and mark all the email on the server as READ. One NICE thing about this is that you can download your email, go offline, read and reply to your email as you wish. When you are ready to send off your replies, just reconnect to the Internet and send off your email. But, even if you don't read all the email on the client and then go back to a different email program like the server-based email programs like Pine or Elm, you won't know which emails were and weren't read. Trust me, this is a pain in the butt.

In Linux, POP3 clients are supported by the in.pop3d daemon and is super simple to install and run. It just loads from /etc/inetd.conf and uses the /etc/passwd or /etc/shadow files to authenticate people.

IMAP4: IMAP4 or Internet Message Access Protocol 4 is the newest email system. Its default method to authenticate users is encrypted BUT you can also add on additional security like have all traffic MD5 encrypted, etc.

Unlike POP3, IMAP4 email clients typically need to be ON-LINE the whole time since you don't download ALL your email at once. The excellent thing about IMAP is that is maintains what emails have been read / not read. So, regardless of the email client you use, you can always read your email easily.

Like I mentioned before, IMAP typically requires the users to be online to read email. I understand that some IMAP4 clients *CAN* download email to be read offline and then re-attach to the mail server and send email and resyncronize what messages have been read/not read. Unfortunately, I don't know of any UNIX clients that can do this. If you know of some, PLEASE LET ME KNOW!

In Linux, IMAP4 clients are supported by the in.imapd daemon and is super simple to install and run. It just loads from /etc/inetd.conf and uses the /etc/passwd or /etc/shadow files to authenticate people.

First, you need to make sure have configured your IPCHAINS or IPFWADM rule sets correctly to allow POP3/IMAP4 traffic and have enabled "in.pop3d" or "in.imapd" in the /etc/inetd.conf file,

Ie, un-# the "pop3d" or "imapd" line in the /etc/inetd.conf file and then run:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}’‘

After that, either/both POP3 and IMAP4 email should work right out of the box.

—– NOTE: When you check your POP-3 email from somewhere on the Internet, your —– username/password are sent in clear text. The same also goes for any other network protocol like TELNET, FTP, etc.

What this means to you is that if someone between your local machines and your POP-3 server is sniffing packets, they will not only be able get your username/password but also get all of your transmitted email too! Now you might be thinking this is paranoid thinking but securing your connections isn't hard and it is better safe than sorry.

So, what can you do to secure these communications? Check out 30 (Section 30) for all the SSH full details!!

NOTE #2: If you allow POP-3 access from anywhere on the Inet, 99% of your users will have trouble SENDING email via SMTP. A few reasons / solutions for this include:

1) They aren't physically connected behind your Linux server. Because of this, your Linux server's SMTP server doesn't want to relay NON-local user email traffic. There is one decent solution to this issue:

Check out the "PopAuth" URL in 5 (Section 5) for full details.

2) Another option to the above issue is to use POP-3 to -SEND- email instead of just receive it. Few POP-3 email clients support this but I know Qualcomm's Eudora supports it fine.

3) The POP-3 client is NOT configured with the "Return Address" as the domain name of your Linux SMTP server.

Finally, if you have multiple Internet email domains (email addresses) running on one Linux server and you want to have different users to be able to send and receive email from the correct email address, etc. Check out the Virtual Email URL in 5 (Section 5)

29 System Backups: Backing up data to HDs, Tape, and floppies

Once you get your system up and running they way you want it, it's only a matter of time before you either make a serious mistake, get HD corruption, or a HD dies all together. COUNT ON IT!

What can you do? Back it up!

So you are probably asking "what should I back up", "how to back things up", etc. Starting out, it's a good idea to backup the STATE of the system onto floppy (or USB flash, etc.). What do I mean by "state"? This small backup will just keep a copy of the primary configuration files, a listing of the binaries installed on your machine, etc. This backup will at least let you get a new system running again with a minor amount of work after re-installing the OS manually. A pain but much better than nothing.

After creating a state config backup, I really recommend need to backup everything. Everything can mean different things to different people. For me, I want a FULL backup where I can restore the entire system onto a new or replacement HD with as little work as possible. To other people, they just want a DATA backup where they just want to backup their various word processing files, pictures, etc. to a safe place.

Both styles of backups can take up a LOT of space which can be a problem. The backup industry used to only have tape drives as the solution. The problem with tape drives is that they can be slow, require multiple tapes, can be very expensive, and unforutunately be unreliable. All of these factors have made hard drive or CD/DVD backups very appealing.

TrinityOS covers backups via:

```
  - STATE backup to a floppy

  - FULL backups to a HD
    * Data being either local to the backup server as well as
     and remote data via NFS / Samba shares

  - Tape backups using the commercial tool Bru for local backups
```

29.1 STATE backups to floppies

Copying files to floppies is EASY. All you need to do is:

- Format the floppy diskette:

mke2fs /dev/fd0

- Mount the floppy

mount -t ext2 /dev/fd0 /mnt/floppy

- Copy at least the following files to the floppy:

Recommended:

- /etc/passwd, /etc/shadow
- /etc/fstab, /etc/raidtab, /etc/inittab
- /etc/lilo.conf, /etc/resolv.conf, /etc/conf.modules, /etc/hosts*
- /var/lib/rpm/fileindex.rpm, /var/lib/rpm/packages.rpm

OPTIONAL (recommended but only if you use these files):

- /etc/smb.conf, /etc/smbpasswd, /etc/smbusers
- /etc/ssh2/*, /etc/dhcpd.conf
- /etc/mail/*

- I would also recommend to record a full file listing of your system as well:

```
ls -laR / | gzip -9 > /mnt/floppy/file-list-‘date +’%b%d’‘.lst.gz
```

- Another GREAT idea comes from the Config-HOWTO to make a backup of your HD's Master Boot Record (MBR). So, instead of manually having to recreate it from your updated details in 4 (Section 4), simply copy the MBR to a file:

Example:

this will backup /dev/hda's table:

```
dd if=/dev/hda of=/boot/mbr.dd bs=512 count=1
cp /boot/mbr.dd of=/mnt/floppy
```

Use this to restore the table:

```
dd if=/mnt/floppy/mbr.dd of=/dev/hda bs=512 count=1
```

You can find more info about the parition table layout at: `<http://www.win.tue.nl/~aeb/partitions/partition_tables-2.html>`

** You will need to redo this backup every time you:

- Add a user
- Change a user's password
- Add/delete any RPMs to your machine
- Make any serious changes to your file system layout

29.2 FULL Backups: local and remote backups using a Hard Drive

Backing systems up to a HD has finally become easy and affordable. Not only are large HDs cheap but you can put them into Firewire/USB enclosures for portability and hot-plug abilities. The same can be said for CD/DVD backups but I find that I /don't/ want to constantly shovel discs in / out and even with compression, backing up 100GB of data requires is a LOT of DVDs.

Here is the TrinityOS "backup-to-disk" script. What this script brings to the table that I haven't seen before is:

- Backs files up to a HD file by file. The backup file is easily restored without having to seek around a massive archive file. My primary goal of this script.
- Network savvy with extensive error checking for network connections
- The script offers real-time logging as well as a copy of the log in the destination backup directory
- The script does extensive error checking (network connectivity, available backup HD space, etc.) so it should tell you why things might not work before you start.
- Extensible to other systems like rsync, cpio, etc.

Please read through the script's comments to understand how it works but here are some highlights:

- This script doesn't tell you how to get NFS or Samba running either on the backup server or backup client. Please read those TrinityOS sections and then come back to this one.
- You must enter in the relevant NFS and Samba info per host (passwords, IPs, etc.) in the configuration section.
- You need to specify all the remote mount points for backup shares as this is how tar works

There are some known limitations with this script that might not work for you. In the future, I plan to make the script support simultaneous NFS backups, use BASH functions, etc.

If you have ideas, URLs for similar backup solutions, or you'd like to see a specific feature added, let me know.

<backup-to-disk START>

```
#!/bin/sh

# TrinityOS HD Backup Script - Supports LOCAL and Remote NFS/SAMBA file systems
#
#  Part of the TrinityOS(tm) documentation
#  Written by David Ranch
#  dranch@trinnet.net

#Version of the TrinityOS backup script
VERSION=v4.8

# v4.8 - 031404 - Initial release on TrinityOS
# v4.7 - 081403 - Added comments to add FULL and differential support
# v4.6 - 050803 - Removed the dos-c volume from dranch-lt-minidock
#               - Added EXTHOST system as some new DNS servers give
#                 hostnames instead of IPs
# v4.5 - 032203 - More comments, moved some things around
# v4.4 - 011603 - made the backup destination more generic
#               - moved away from hosts being IPs back to names.  Very
#                 ugly and the df issue was due to line wrapping
#               - Added the compression of the log files (10.5M to 1M)
# v4.3 - 011003 - Added verbage when NFS mount checks hang
#               - Attempt to use df -P to fix parsing problems
# v4.2 - 122602 - Moved to using IP addresses vs. hostnames to help with
#                 df parsing issues
# v4.1 - 122402 - Updated Trinity directory list
#               - removed a lame if/then that would stop remounting NFS
#                 if ANY nfs mounts existed for that specific remote client
#                 UNFORTUNATELY, Linux will allow duplicate NFS mounts...
# v4.0 - 112802 - Fixed the estimation phase for Samba clients
# v3.9 - 112502 - Added the backup of the backup-to-disk to the dest disk
# v3.8 - 090602 - Corrected the estimated backup size for local backups
# v3.7 - 090602 - Added additional text for firewall situations
#               - moved -check termination point
# v3.6 - 090402 - Added additional formatting to improve backup output
#               - Removed unneeded souce backup estimation
#               - Added the ability to disable file-by-file logging
#               - Changed the colors of the backup window
#               - added the "check" option to check for minimium disk space
# v3.5 - 090302 - Added more FSs on Trinity
# v3.4 - 070702 - Added the spawning of a logging window
#               - Added more comments
# v3.3 - 061802 - Added some more comments
# v3.2 - 060102 - Fixed some tail information errors
#               - Deleted the use of restarting CRON as it is already dynamic
#               - Fixed the problem where NFS couldn't umount at the end
# v3.1 - 053002 - Added some more comments
# v3.0 - 040202 - changes some mount points, more formatting, etc.
# v2.9 - 031902 - fixed the BACKUPPATH for Trinity to watch for sub-mounted dirs
```

```
# v2.8 - Added the capture of an error log
# v2.7 - Added addition error checking, more debug statements, etc.
# v2.6 - Only backup one physical FS at a time
# v2.5 - Added compression and HOT backups
# v2.4 - added Samba support
# v2.3 - Fixed backup paths to be more normal instead of overly nested
# v2.2 - added support for multiple NFS mountpoints
# v2.1 - changed to backup machine at home with additional testing
# v2.0 - added lots of network availablity testing
# v1.0 - Initial version


#NOTES
#-----
#  - This backup script is intended to be run on the backup SERVER and not on
#    the backup CLIENT
#
#  - For remote NFS backups, the backup client needs to be the NFS server.
#    The backup server is only an NFS client.
#
#  - Remote backups are done using RELATIVE domain names.  ie host names like
#     "roadrunner" vs. "roadrunner.acme123.com".  If you cannot ping just the hostname
#     from the backup server, you need to fix this via the /etc/resolv.conf file
#
#
#  - NFS users:
#
#    No need to check if CDROMS are mounted on the client as they are seperate
#    file systems that are not exported to NFS.  If they are exported, just make sure
#    they aren't included in the BACKUPPATH variable below
#
#        This does NOT apply to backups via SMB !!
#
#
#  - Samba users
#
#      Nothing has to be loaded for things to work properly
#
#
#  - Compression
#
#      Compression isn't currently functional.  Id like to do this via one pass
#      but I don't see how that will be possible with using TAR
#
#
#  - Seti
#
#      This script looks to see if the Seti program is running.  If you arent
#      running seti or dont know what it is, dont worry about it.
#
```

```
# TO DO
# -----
#
# 1. Re-write the script to exensively use Bash functions instead.  Put the
#    unmounting into a function so when -check is used, it cleans up
#
# 2. update the logic to avoid duplicate NFS mounts
#
# 3. run a check to make sure the partition table and MBR are imaged
#
# 4. make the script multi-instance aware so if say multiple NFS backups are
#    running, additional run scripts won't clobber the first run NFS backup
#
# 5. add command line support for FULL vs. DIFFERENTIAL support


#HOW TO USE THIS SCRIPT
#----------------------
#
# 1. Edit the BACKUP variables below to reflect the desired CLIENT machines,
#     method for backup, etc.
#
#
# 2. Mount the local BACKUP disk
#
#      For example:
#
#         IDE BUS:       mount /dev/hdc1 /mnt/backup-disk
#
#         FireWire BUS:  mount /dev/sdd1 /mnt/backup-disk
#
#
#    -----------------------------------------------------------------------
#    NOTE:  if the file "/mnt/backup-disk/backup-drive-ready" doesn't exist
#           on the backup drive, the backup will abort.  This is just to make
#           sure that not just any HD will be used for the backup
#    -----------------------------------------------------------------------
#
#
# 3. NFS Users: Start up **REMOTE** NFS daemons
#
#       This is not needed for LOCAL or SMB backups
#
#   LOCAL:  start the NFS client  (OPTIONAL as this is done automatically)
#       /etc/rc.d/init.d/portmap start
#
#
#   REMOTE: start the NFS server
```

```
#
#       /etc/rc.d/init.d/portmap start
#       /etc/rc.d/init.d/nfs start
#
#          NOTE #1: make sure that the backup clients IP addr is in
#                    its /etc/exports file
#
#          NOTE #2: some hosts might need their IPCHAINS/IPTABLES
#                    firewall removed before NFS will work
#
#
# 4. Delete old CLIENT data directory on /mnt/backup-disk
#
# 5. Start new backup by running this script with the given host:
#
#        ./backup-to-disk coyote
#
#     You can also run "./backup-to-disk coyote -check"
#       to understand the backup requirements (runs the estimation
#       phase and then exits.
#



#Setup the BACKUP variables
#-------------------------------------------------------------------------------------

clear


if [ "$1" == "" ]; then
   echo -e "\n\n** ERROR **: Backup source not specified "
   echo -e "\nbackup-to-disk usage:  \n"
   echo -e "  backup-to-disk < roadrunner | coyote | wile | acme > <-check>"
   echo -e "\n      -check : determine client disk requirements then exit\n\n"
   exit 1
fi

case $1 in

  roadrunner)
  # Backup via NFS
      #How to back things up
        BACKUPMETHOD=NFS
      #The machine to be backed up
        CLIENT=roadrunner
      #Backup SOURCE on the REMOTE machine
        SOURCEMOUNT="/mnt/nfs"
      #What files are being backed up from the SOURCE
        MOUNTLIST="/ /var /home/johndoe /home/johndoe/pictures /home/johndoe/movies /tmp"
```

```
        UNMOUNTLIST="/tmp /home/johndoemovies /home/johndoe/pictures /home/johndoe /var /"
      #Backup Path
        BACKUPPATH="bin boot bru dev dosc etc home home/johndoe \
home/johndoe/pictures home/johndoe/movies lib misc mnt opt root sbin tmp usr var"
      #Backup destination
        BACKUPDEST="/mnt/backup-disk"
        DEST_PATH="/mnt/backup-disk"
      #Do we want to do compression
        COMPRESSION=no
      #Backup options for NFS
        NFSOPTIONS="rsize=8192,wsize=8192"
      #Enable logging of every backed up file to output file
        LOGGING=yes
     ;;

  coyote)
  #Backup via Samba
      #How to back things up
        BACKUPMETHOD=SAMBA
      #The machine to be backed up
        #  SAMBA wants short names (NetBIOS)
        CLIENT=coyote
      #Backup SOURCE on the REMOTE machine
        SOURCEMOUNT="/mnt/samba"
      #What files are being backed up from the SOURCE
        MOUNTLIST="coyote-c coyote-d"
        UNMOUNTLIST="coyote-d coyote-c"
      #Backup Path
        BACKUPPATH="coyote-c coyote-d"
      #Backup destination
        BACKUPDEST="/mnt/backup-disk"
        DEST_PATH="/mnt/backup-disk"
      #Do we want to do compression
        COMPRESSION=no
      #Backup options for SAMBA
        SMBOPTIONS="username=johndoe,password=<your-password-here>"
      #Enable logging of every backed up file to output file
        LOGGING=yes
     ;;

  wile|wilee)
  #Backup via local
      #How to back things up
      BACKUPMETHOD=LOCAL
      #The machine to be backed up
        CLIENT=wile
      #Backup SOURCE on the LOCAL machine
        SOURCEMOUNT="/"
      #What files are being backed up from the SOURCE
        MOUNTLIST=""
```

```
        UNMOUNTLIST=""
      #Backup Path
        LOCALMOUNT="/dev/sdb3 /dev/sdc2 /dev/sda1 /dev/sdb1 /dev/sdc1"
        BACKUPPATH="/ /usr/src /mnt/dos-c /mnt/dos-d /mnt/dos-e"
      #Backup destination
        BACKUPDEST="/mnt/backup-disk"
        DEST_PATH="/mnt/backup-disk"
      #Do we want to do compression
        COMPRESSION=no
      #Enable logging of every backed up file to output file
        LOGGING=yes
     ;;

  acme | acme-corp)
  # Backup via NFS
      #How to back things up
        BACKUPMETHOD=NFS
      #The machine to be backed up -- USE IP address to avoid "df" parsing iss.
        CLIENT=acme
      #Backup SOURCE on the REMOTE machine
        SOURCEMOUNT="/mnt/nfs"
      #What files are being backed up from the SOURCE
        MOUNTLIST="/"
        UNMOUNTLIST="/"
      #Backup Path
        BACKUPPATH="/"
      #Backup destination
        BACKUPDEST="/mnt/backup-disk"
        DEST_PATH="/mnt/backup-disk"
      #Do we want to do compression
        COMPRESSION=no
      #Backup options for NFS
        NFSOPTIONS="rsize=8192,wsize=8192"
      #Enable logging of every backed up file to output file
        LOGGING=yes
   ;;

   -h)
     echo -e "\n\n    ** ERROR:  Hostname $1 not recognized.  Aborting\n\n."
     exit 1
   ;;

   *)
     echo -e "\n\n    ** ERROR:  Hostname $1 not recognized.\n"
     echo -e "Usage:  \n"
     echo -e "        backup-to-disk \[roadrunner | coyote | wile | acme\]\
 <-check>\n"
     echo -e "        -check - calculates requires disk for remote host and exits\n\n"
     exit 1
   ;;
```

```
esac


#LOCAL machine’s network interface name
EXTIF=eth0





#----------------------------------------------------------------------------------
#-- DO NOT EDIT BELOW THIS LINE UNLESS YOU KNOW WHAT YOU ARE DOING ----------------
#----------------------------------------------------------------------------------

echo -e "\nWelcome to the TrinityOS HD backup script $VERSION"
echo -e "------------------------------------------------\n\n"


#Calculate the SERVERs IP address
#
EXTIP=‘/sbin/ifconfig | grep -A 4 $EXTIF | awk ’/inet/ { print $2 } ’ \
| sed -e s/addr://‘
EXTHOST=‘host $EXTIP | awk ’{print $5}’‘


#Backup DESTINATION on the LOCAL machine - should be a LARGE disk
DEST_DIR="‘date "+%m%d%y"‘"
DEST="$DEST_PATH/$CLIENT-$DEST_DIR"

#Automatic backup time determination - do not edit
START=‘date‘


if [ "$LOGGING" == "yes" ]; then
   #Override the variable contents now with the logging destination
   LOGGING="$DEST/$CLIENT-backup.log"
  else
   LOGGING="/dev/null"
fi

if [ "$BACKUPMETHOD" == "NFS" ]; then
  echo -e "\nMake sure that you have enabled the following on [ $CLIENT ] \n"
  echo -e "echo 262144 > /proc/sys/net/core/rmem_default"
  echo -e "echo 262144 > /proc/sys/net/core/rmem_max\n\n"
  echo -e "\nPAUSING for 10 seconds\n"
  sleep 10
fi

if [ "$BACKUPMETHOD" == "SAMBA" ]; then
  echo -e "\nMake sure that you have disabled any Anti-Virus softeware on the backup"
```

```
  echo -e "source.   If you don’t do this, the remote system can and will do wierd"
  echo -e "things such as report file size changes during backup, etc."
  echo -e "\nPAUSING for 10 seconds\n"
  sleep 10
fi





#If we are using compression, make sure that Seti is NOT running
if [ "$COMPRESSION" == "yes" ]; then
  GZIP="z"
  if [ -f /usr/local/sbin/start-seti ]; then
     SETIPID=‘ps ax | grep "seti" | grep -v "grep" | awk ’{print $1}’‘
     if [ -n "$SETIPID" ]; then
        echo -e "    ** Stopping SETI.."
        kill $SETIPID
        SETIPID=‘ps ax | grep "seti" | grep -v "grep" | awk ’{print $1}’‘
        if [ -n "$SETIPID" ]; then
          echo -e "      ** ERROR: Could not stop SETI"
          exit 1
        fi
     fi
     mv /etc/cron.hourly/start-seti /etc/cron.hourly.disabled/
     echo -e "      ** Warning: Restarting cron to then disable seti from starting"
     /etc/rc.d/init.d/crond restart
  fi
 else
  GZIP=""
fi





echo -e "\nPreparing to backup [ $CLIENT ] to [ $EXTIP ] via [ $BACKUPMETHOD ]"


if [ "$BACKUPMETHOD" == "NFS" ] || [ "$BACKUPMETHOD" == "SAMBA" ]; then
   # Verify the required NETWORK subsystem is running..

   if [ ! -n "‘ping -c 1 $CLIENT | grep icmp_seq‘" ]; then
     echo -n "      ** ERROR - ICMP: Cannot reach $CLIENT  Aborting.\n\n"
     exit 1
   fi
   echo -e "\n    ICMP: [ $CLIENT ] is reachable.."
fi


#Do tests based upon the backup method
#
```

```
if [ "$BACKUPMETHOD" == "NFS" ]; then
  echo -e "    NFS:  checking PORTMAP.."
  if [ ! -n "‘ps ax | grep portmap | grep -v "grep portmap"‘" ]; then
    echo -e "\n    WARNING - NFS: PORTMAP not running.  Attempting to start it.."
    /etc/rc.d/init.d/portmap start
    echo -e "\n"
    if [ ! -n "‘ps ax | grep portmap | grep -v "grep portmap"‘" ]; then
      echo -e "      ** ERROR - NFS: Could NOT start PORTMAP.  Aborting."
      exit 1
    fi
  fi
  echo -e "    NFS:  PORTMAP is running.."


  echo -e "    NFS:  checking exports [any hangs at this point are due to remote FWs]"
  echo -e "          or the remote host is not running NFS at this point"

  #Make sure we can mount the remote machine
  #
  # Newer NFS servers export the DNS name and not the IP
  #
  if [ -z "‘showmount -e $CLIENT | grep "/" | awk ’{print $2}’ | grep "$EXTHOST"‘" ] -o \
[ -z "‘showmount -e $CLIENT | grep "/" | awk ’{print $2}’ | grep "$EXTIP"‘" ]; then
    echo -e "\n    ** ERROR - NFS: Local machine not in $CLIENT export list.  Aborting."
    echo -e "\nExports list was:"
    echo -e "----------------"
    showmount -e $CLIENT
    echo -e "----------------"
    echo -e "\nExpected EXPORTed IP: $EXTIP (old NFS servers)"
    echo -e "Expected EXPORTed DNS name: $EXTHOST (new NFS servers)"
    exit 1
  fi
  echo -e "    NFS:  Remote machine [ $CLIENT ] is properly exporting to our IP"


  echo -e "    NFS:  Starting to run NFS mounts.."
  #Mounting the remote file systems

# BUG:
#
#   WRONG:  Linux allows duplicate NFS mounts, fix this logic to test for
#           each specific mount
#
#  if [ ! -n "‘df | grep $CLIENT‘" ]; then
    echo -e "    NFS:  Mounting [ $CLIENT ] with options: [ $NFSOPTIONS ]"

    for I in $MOUNTLIST; do
      echo "      Mounting: [ $SOURCEMOUNT$I ] "
      mount -t nfs -o $NFSOPTIONS $CLIENT:$I $SOURCEMOUNT$I
    done
```

```
    if [ ! -n "‘df | grep $CLIENT‘" ]; then
      echo -e "      ** ERROR - NFS:  Cound not mount [ $CLIENT ]"
      exit 1
    fi
#  fi
  echo -e "    NFS:  [ $CLIENT ] successfully mounted."
fi


if [ "$BACKUPMETHOD" == "SAMBA" ]; then
  echo "    SMB:  Checking status of remote SMB host.."
  #Make sure that the remote machine is responding to SAMBA requests
  if [ -z "‘smbclient -L //$CLIENT -N | grep -i "disk"‘" ]; then
     echo -e "      ** ERROR:  [ $CLIENT ] is not responding to SAMBA requests"
     exit 1
  fi
  echo "      Host [ $CLIENT ] is reponding to SMB requests.."

  #Samba - Mount things up
  echo -e "    SMB:  Starting to run SMB mounts.."
  for I in $MOUNTLIST; do
    if [ ! -d $SOURCEMOUNT/$I ]; then
       echo -e "      ** ERROR:  destination mount [ $SOURCEMOUNT/$I ] point does not exist"
       exit 1
    fi
    echo "        [ $I ] mount point already exists.  Continuing.."

    if [ -z "‘df | grep $I‘" ]; then
       echo "          Mounting: [ $I ]"
       echo "        Mounting [ $SOURCEMOUNT/$I ] - Please provide required passwords"
       /usr/bin/smbmount //$CLIENT/$I $SOURCEMOUNT/$I -o $SMBOPTIONS
      else
       echo "        Samba mount [ $I ] already mounted.  Continuing.."
    fi
  done

  if [ ! -n "‘df | grep $CLIENT‘" ]; then
    echo -e "      ** ERROR - SAMBA:  Cound not mount [ $CLIENT ]"
    exit 1
  fi
  echo -e "    SAMBA: [ $CLIENT ] successfully mounted."
fi


# Must run this AFTER the network is up to get CLIENT info
#

#Is the backup media really present
#  This looks for a file called "backup-drive-ready" on the backup DESTINATION
```

```
#
if [ ! -f $BACKUPDEST/backup-drive-ready ]; then
   echo -e "\n    ** ERROR **  Backup media isn’t present.  Make sure the dest \
backup drive"
   echo -e "                 is installed and mounted.\n"
   echo -e "                 If the media IS mounted properly, make sure the file"
   echo -e "                 $BACKUPDESK/backup-drive-ready exists.  Until then..\n\n"
   echo -e "                 Aborting.\n\n"
   exit 1
fi
echo -e "\n    Backup destination media is present"


#Does the backup destination have enough space?

#How big is the REMOTE backup
if [ "$BACKUPMETHOD" == "NFS" ] || [ "$BACKUPMETHOD" == "SAMBA" ]; then
  TOTAL=0

  # The issue must be the use of the ":"
  #    #coyote wants awk var3 and not var2
  #    #roadrunner needs awk var2

  #coyote
  #coyote:/  18951536  11212792   6776048  62% /mnt/nfs

  #roadrunner
  #

  #acme
  #//acme/acme-c
  #                    2096832   1974688    122144  94% /mnt/samba/acme-c

  if [ "$BACKUPMETHOD" == "SAMBA" ]; then

     #Samba’s use of screws up awk
     echo "    Calcing Samba size"
     for I in ‘df -P | grep "$SOURCEMOUNT" | awk ’{print $3}’‘; do
     TOTAL=$(($TOTAL + $I))
     done

    else

     echo -e "\n    Calcing NFS size"

     # 122502 - moving from $3 to $2 though I dont know why
     #          -- maybe something in the src nfs hostname

     # awk-3 is good for coyote
     # awk-2 is good for roadrunner
```

```
        # must change this to do it via the mount point and not the sourcemount
        # parse for /mnt/nfs/dos-c and not 192.168.0.7:/dos-c
        # because the lines wrap on long lines.  also use df -Tk to
        # help parsing
        #
        # I need to either parse from the RIGHT to the left or use
        # some other feature of awk

     for I in `df -Pk | grep "$SOURCEMOUNT" | awk '{print $3}'`; do
     TOTAL=$(($TOTAL + $I))
     done
  fi

  echo -e "\n    ESTIMATED Backup size      : $TOTAL"
  BACKUPDESTDU=`df -Pk | grep $BACKUPDEST | awk '{print $4}'`
  echo "    Backup DESTINATION capacity: $BACKUPDESTDU"
fi


#How big is the LOCAL backup
if [ "$BACKUPMETHOD" == "LOCAL" ]; then
  TOTAL=0
  for I in $LOCALMOUNT; do
    #acme
    #/dev/sdb3              7302300   2240072   4691284  32% /

    J=`df -P | grep "$I" | awk '{print $3}'`
    echo "      - Checking mount: $I - SIZE: $J"
    TOTAL=$(($TOTAL + $J))
  done
  echo -e "\n    ESTIMATED Backup size      : $TOTAL"

  BACKUPDESTDU=`df -P | grep $BACKUPDEST | awk '{print $4}'`
  echo "    Backup DESTINATION capacity: $BACKUPDESTDU"
fi

if [ $TOTAL -ge $BACKUPDESTDU ]; then
  echo -e "\n   ** ERROR **  NOT ENOUGH DISK SPACE on backup device. Aborting.\n\n"
  exit 1
fi
echo -e "\n      [ $BACKUPDEST ] has enough diskspace to backup host [ $CLIENT ]"


if [ "$2" = "-check" ]; then
  echo -e "\n*******************************************************"
  echo -e "** ABORT:                                            **"
  echo -e "**                                                   **"
  echo -e "**    -check command line option specified.  Exiting. **"
  echo -e "*****************************************************\n\n"
```

```
  exit 0
fi


echo -e "\n    Backup Destination is: [ $DEST ] "
mkdir $DEST > /dev/null
if [ ! -d $DEST ]; then
  echo "    ** ERROR:  Could not create destination directory"
  exit 1
fi
echo "      Created the destination directory.."


#Get the backup size - dont use -c but use -s instead since you will
#  match on multiple "total" lines
#
echo -e "\n---------------------------------------------------------------------" \
> $DEST/$CLIENT-backup.log
echo -e "Auto-generated by the TrinityOS backup script $VERSION" >> $DEST/$CLIENT-backup.log
echo -e "\nThis is a FULL backup of host: $CLIENT" >> $DEST/$CLIENT-backup.log
echo -e "\nRun from machine: `uname -a`" >> $DEST/$CLIENT-backup.log
echo -e "\nBackup START: $START" >> $DEST/$CLIENT-backup.log

echo "      ESTIMATED backup size: $TOTAL"
echo -e "\nESTIMATED backup size: $TOTAL" >> $DEST/$CLIENT-backup.log


# This section is not required as the $TOTAL calculation above is accurate enough
#
#   THIS SECTION WILL BE REMOVED SHORTLY
#
#if [ "$BACKUPMETHOD" == "LOCAL" ]; then
#   #Calc space for local volumes since du does't do what we expect
#   CALCEDSIZE=0
#   echo "    Calculating actual backup space requirements.  Please wait."
#   for I in $BACKUPPATH; do
#     J=`du -s -x $I | awk '{print $1}'`
#     #echo "$I"
#     CALCEDSIZE=$(($CALCEDSIZE + $J))
#   done
#   echo "    Initial backup size: $CALCEDSIZE"
#   echo -e "\nINITIAL backup size: $CALCEDSIZE" >> $DEST/$CLIENT-backup.log
# else
#   #Calc space for NFS and SMB
#   echo "    Calculating actual backup space requirements.  Please wait.."
#   CALCEDSIZE="`du -s --exclude /mnt/mnt $SOURCEMOUNT | awk '{print $1}'`"
#   echo -e "\n    Calculated backup size: $CALCEDSIZE"
#   echo -e "\nCalculated backup size: $CALCEDSIZE" >> $DEST/$CLIENT-backup.log
#fi
```

```
if [ "$BACKUPMETHOD" == "NFS" ]; then
  #Create placeholder dirs
  mkdir -p $DEST/mnt/floppy > /dev/null
  mkdir -p $DEST/mnt/cdrom > /dev/null
  mkdir -p $DEST/lost+found > /dev/null
  mkdir -p $DEST/proc > /dev/null
fi

#Put of a copy of the backup script on the backup drive
cp /root/backup-to-disk $DEST/backup-to-disk

echo -e "\n\nSpawning logging window..\n"
/usr/X11R6/bin/xterm -fg white -bg darkblue -title "$CLIENT backup-to-disk=log-window" \
    -e tail -f $DEST/$CLIENT-backup.log &

echo -e "\nBacking up data on host $CLIENT with permissions, ownerships, etc"
echo -e "=============================================================================="
echo -e "\n\n-------------------------------------------------------------------------------"
echo -e "Full backup logs can be monitored by running:\n"
echo -e "     tail -f $DEST/$CLIENT-backup.log"
echo -e "\n-------------------------------------------------------------------------------\n\n"
echo -e "\n-------------------------------------------------------------------------------" >> $DEST/$CLIENT-back
echo -e "Full backup logs can be monitored by running:\n" >> $DEST/$CLIENT-backup.log
echo -e "     tail -f /mnt/$BACKUPDEST/$CLIENT-backup.log" >> $DEST/$CLIENT-backup.log
echo -e "\n-------------------------------------------------------------------------------" >> $DEST/$CLIENT-back

for I in $BACKUPPATH; do
  echo -e "\n---------------------------------------------------"
  echo -e "Messages below are due to ERRORS encountered during"
  echo -e "the backup:"
  echo -e "---------------------------------------------------"

  echo -e "\n------------------------------------------------------" >> $DEST/$CLIENT-backup.log
  echo -e "Messages below are due to ERRORS encountered during" >> $DEST/$CLIENT-backup.log
  echo -e "the backup:" >> $DEST/$CLIENT-backup.log
  echo -e "------------------------------------------------------" >> $DEST/$CLIENT-backup.log

  echo -e "Backing up     : [ $I ]\n"
  echo -e "Backing up     : [ $I ]\n" >> $DEST/$CLIENT-backup.log
  #do this manually to not create bakups with /mnt/mnt/backup/mnt/nfs/bin
  cd $SOURCEMOUNT/$I
  mkdir $DEST/$I > /dev/null
  if [ ! -d $DEST/$I ]; then
    echo "      ** ERROR:  Could not create destination directory"
    exit 1
  fi

  # ***  HEAVY LIFTING  ***
  #
  #tar cpsf - $SOURCEMOUNT/$I | (cd $DEST; tar xvpvf - )
```

```
  #Be sure to NOT to backup anything other than the local filesystem
  tar clpsf - . | (cd $DEST/$I; tar xpvf - ) 2>> $DEST/$CLIENT-backup-errs.log >> $LOGGING

  echo -e "DONE backing up: $I"
  echo -e "DONE backing up: $I" >> $DEST/$CLIENT-backup.log
  echo -e "------------------------------------------------------"
  echo -e "------------------------------------------------------" >> $DEST/$CLIENT-backup.log
done


echo -e "\n\n==============================================================================="
echo -e "\n\n===============================================================================" \
  >> $DEST/$CLIENT-backup.log
echo -e "Backup COMPLETED.\n\n"
echo -e "Backup COMPLETED.\n\n" >> $DEST/$CLIENT-backup.log


#Get the final backup size - dont use -c but use -s instead since you will
#  match on multiple "total" lines
#
echo "Calculating FINAL backup size.. [ please wait.. ]"
echo "Calculating FINAL backup size.. [ please wait.. ]" >> $DEST/$CLIENT-backup.log
CLOSING=‘du -s $DEST | awk ’{print $1}’‘
echo -e "    ESTIMATED backup size: $TOTAL"
echo -e "    ESTIMATED backup size: $TOTAL" >> $DEST/$CLIENT-backup.log
echo -e "    FINAL backup size    : $CLOSING"
echo -e "    FINAL backup size    : $CLOSING" >> $DEST/$CLIENT-backup.log

#get out of any existing NFS/SAMBA partions
cd /root


if [ "$BACKUPMETHOD" == "NFS" ] || [ "$BACKUPMETHOD" == "SAMBA" ]; then
  echo -e "\nUnmounting [ $CLIENT ] "
  for I in $UNMOUNTLIST; do
    echo "      UNMounting: [ $SOURCEMOUNT/$I ] "
    umount $SOURCEMOUNT/$I
  done
fi


if [ "$BACKUPMETHOD" == "NFS" ]; then
  echo -e "\nUnloading PORTMAP"
  /etc/rc.d/init.d/portmap stop

  if [ -n "‘ps ax | grep "portmap" | grep -v "grep portmap"‘" ]; then
    echo -n "\nCould NOT stop PORTMAP.  Aborting."
    exit 1
  fi
fi
```

```
#If we were using compression and seti is on this machine, restart it
if [ "$COMPRESSION" == "yes" ]; then
  if [ -f /usr/local/sbin/start-seti ]; then
     echo -e "    ** Starting SETI.."
     /usr/local/sbin/start-seti
     SETIPID=‘ps ax | grep "seti" | grep -v "grep" | awk ’{print $1}’‘
     if [ -z "$SETIPID" ]; then
        echo -e "      ** ERROR: Could not start SETI"
        exit 1
     fi
     mv /etc/cron.hourly.disabled/start-seti /etc/cron.hourly
  fi
fi

#WILL BE REMOVED
#tail --lines 16  $DEST/$CLIENT-backup.log

echo -e "\nBackup STARTed: $START"
echo -e "\nBackup STARTed: $START" >> $DEST/$CLIENT-backup.log
echo -e "Backup STOPped: ‘date‘\n\n"
echo -e "Backup STOPped: ‘date‘\n\n" >> $DEST/$CLIENT-backup.log

if [ "$LOGGING" != "/dev/null" ]; then
   echo -e "Compressing all log files"
   gzip -9 $DEST/$CLIENT-backup.log
   gzip -9 $DEST/$CLIENT-backup-errs.log
  else
   echo -e "Logging NOT enabled.  Log Compression stopped."
fi

echo -e "\nEnd of TrinityOS HD backup script $VERSION"
echo -e "==============================================================================\n\n"
```

<backup-to-disk STOP>

To get the script, download it from the TrinityOS-archives.tar.gz file on Dranch's web site. PLEASE, don't try to cut and paste this into a new file:

`<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>`

Once you have the script, put it in the ROOT user's directory. Why root? Well, you'll need to be root to mount the remote or local backup HD. You'll need to be root to backup all the local file systems. Etc.

To make it executable, run:

```
    chmod 700 /root/backup-to-disk
```

To run it, simply type something like:

```
   /root/backup-to-disk coyote
```

29.3 Full backups using a Tape drive:

```
+---------------------------------------------------------------------------+
| //// Prerequisites: \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
+---------------------------------------------------------------------------+
|                                                                           |
|   + Bru (tape software is installed).  Check by using this command:       |
|                                                                           |
|       whereis bru                                                         |
|                                                                           |
|                                                                           |
|   + Compiled a kernel to either support (at MINIMUM).  Please see the     |
|     Kernel Compiling Section for more details on how to do the following: |
|                                                                           |
|   * IDE tape drives                                                       |
|                                                                           |
|   Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE) |
|   Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE)                 |
|                                                                           |
|              or                                                           |
|                                                                           |
|   * your specific SCSI controller with SCSI tape support                  |
|                                                                           |
|     SCSI support (CONFIG_SCSI)                                            |
|     SCSI tape support (CONFIG_CHR_DEV_ST)                                 |
|     Verbose SCSI error reporting (kernel size +=12K) (CONFIG_SCSI_CONSTANTS)|
|                                                                           |
|   .....and for example, the Adaptec 1522 SCSI controller:                 |
|   Adaptec AHA152X/2825 support (CONFIG_SCSI_AHA152X)                      |
|                                                                           |
|                                                                           |
|   + A properly installed IDE (master/slave) or a SCSI tape drive          |
|     (with proper SCSI IDs and termination)                                |
|                                                                           |
|                                                                           |
|   + Files created/edited:                                                 |
|                                                                           |
|       /usr/local/sbin/bru-fullbackup                                      |
|       /etc/brutab                                                         |
|       /etc/bruxpa                                                         |
|                                                                           |
+---------------------------------------------------------------------------+
```

(Bru isn't free if you don't install Redhat or Caldera but it's the best Linux backup software out there. This is one place you just CAN'T skimp!) If you don't want to use Bru, at least use CPIO instead of TAR. Tar does work fine UNTIL you hit an error on the tape. After that, tar will shutdown and you'll be screwed since it can't do data recovery. CPIO on the other hand can at least skip the bad file.

NOTE: I've noticed that the behavior of BRU between v14.3 and 15.0 (Bru2000) is quite different. Still works though!)

```
                +-------------------------------------------------------+
```

```
| All the BRU documentation is available at:          |
|                                                     |
|         http://www.estinc.com/brumanual/toc.html    |
+-----------------------------------------------------+
```

NOTE: This is ONLY for users running anything LESS than Glibc-2.0.7-19:

- To check , run "rpm -q glibc"

- Edit /etc/profile and add your appropriate time zone above the "export" command (this is for the Pacific time zone):

TZ=PDT

Next, find the line that starts with "export" and add "TZ" to the end of it. Here is my "export" line:

export PATH PS1 HOSTNAME HISTSIZE HISTFILESIZE USER LOGNAME MAIL NNTPSERVER TZ

Next, you need to setup BRU to understand your tape drive. Personally, I would recommend to use ESTINC's setups at:

`<http://www.estinc.com/brutabs.html>`

Or, startup Xwindows and run "bruconfig" and configure it this way.

```
--< /etc/brutab START>--
# BRUTAB Globals
#+MAXWRITES=1000
#+RAWZBUFSIZE=500
#+RECYCLEDAYS=0
#+OVERWRITEPROTECT=YES
#+ZBUFSIZE=5M
#
# Changed Zbufsize from 500k to 2M
# Changes size from 4000MT to 8000MT
# Changed bufsize from 32k to 64k

#### NOTE!!!  BRU tracks the size of uncompressed files by design.
####
####          So, when using either software or hardware compression, simply set
####              the tape drive capacity size to ZERO in /etc/brutab (size=0).

# Devices
/dev/st0 devname="NS-8 Drive, 8GB, rewind" \
     size=0MT bufsize=16k \
     shmseg=10 shmmax=200k \
     rawtape tape shmcopy rewind autoscan \
     fmtcmd="mt -f /dev/st0 erase" \
     rfmcmd="mt -f /dev/st0 fsf" \
     bfmcmd="mt -f /dev/st0 bsf" \
     retencmd="mt -f /dev/st0 reten" \
     rewindcmd="mt -f /dev/st0 rewind" \
     eodcmd="mt -f /dev/st0 seod" \

/dev/nst0 devname="NS-8 Drive, 4GB, norewind" \
```

```
     size=0MT bufsize=16k \
     shmseg=10 shmmax=200k \
     rawtape tape shmcopy norewind noautoscan # # # # # # \
     fmtcmd="mt -f /dev/st0 erase" \
     rfmcmd="mt -f /dev/nst0 fsf 1" \
     bfmcmd="mt -f /dev/nst0 bsf 1" \
     retencmd="mt -f /dev/st0 retension" \
     rewindcmd="mt -f /dev/st0 rewind" \
     eodcmd="mt -f /dev/nst0 eod" \

# /dev/null device, useful for testing
/dev/null devname="Bit Bucket" \
     size=0 bufsize=20k \
     norewind noautoscan

- devname="stdin/stdout" \
     size=0 bufsize=20k \
     norewind noautoscan

--< /etc/brutab END>--
```

Now we need to setup an exclude file so you don't backup things like CD-ROM drives or compress ZIP files, etc. First, backup the original file by doing "mv /etc/bruxpat /etc/bruxpat.orig" and then create this file and edit it to fit your needs:

```
--< /etc/bruxpat Start>--
# Updated 03/09/99 to change the tape drive capacity to "0" for compression reasons
# Updated 11/25/98 to add no compression of RAR files --dranch
# Updated 7/23/98 to add Cdrom2-8 exclusion  --dranch
# Updated 6/14/98 to add [aA] for the ARJ multivolume stuff --dranch
#
# This file is used by -X option to provide an inclusion/exclusion
# list.  For each pathname of a file selected for backup, each line
# of this file is examined for a pattern, and that pattern is applied
# to the pathname.  If the pattern matches, the appropriate action
# is taken (the pathname is accepted or rejected).  If the pathname
# makes it through all the patterns it is accepted.
#
# These patterns will ONLY be applied to filenames that are part
# of directories that are specified on the bru command line (or
# the current directory, if none are specified).
#
#
# Each command line in the bruxpat file (the file you are now reading)
# consists of a control field and a pattern.  The pattern
# is separated from the control field by whitespace.  Control field
# characters are:
#
#       i       Include this pathname if pattern matches.  The
#               pathname is accepted and no further patterns are
#               applied.
```

```
#                                *** NOTE ****
#               It stops trying on the first pattern match found
#               and passes the filename.  Since it scans patterns
#               in the order listed, "include" patterns normally
#               should be listed before any "exclude" patterns.
#
#       x       Exclude this pathname if pattern matches.  The
#               pathname is rejected and no further patterns are
#               applied.
#
#       z       Exclude this pathname from compression if pattern
#               matches (if the -Z option is specified).
#
#       s       The pattern is a shell style wildcard pattern except
#               that '/' characters are not treated as special characters.
#
#       r       The pattern is a regular expression (same as used by the "grep"
#               command).
#
#       l       The pattern is a literal string.
#

# Exclude all core files
xs      */core
xs      core

# Don't try to get the stuff in /proc
xs      /proc/*
xs      ./proc/*

# Don't backup the CD-Rom
xs      /home/hpe/CDROMs/Cdrom0/*
xs      ./home/hpe/CDROMs/Cdrom0/*
xs      /home/hpe/CDROMs/Cdrom1/*
xs      ./home/hpe/CDROMs/Cdrom1/*
xs      /home/hpe/CDROMs/Cdrom2/*
xs      ./home/hpe/CDROMs/Cdrom2/*
xs      /home/hpe/CDROMs/Cdrom2/*
xs      ./home/hpe/CDROMs/Cdrom2/*
xs      /home/hpe/CDROMs/Cdrom3/*
xs      ./home/hpe/CDROMs/Cdrom3/*
xs      /home/hpe/CDROMs/Cdrom4/*
xs      ./home/hpe/CDROMs/Cdrom4/*
xs      /home/hpe/CDROMs/Cdrom5/*
xs      ./home/hpe/CDROMs/Cdrom5/*
xs      /home/hpe/CDROMs/Cdrom6/*
xs      ./home/hpe/CDROMs/Cdrom6/*
xs      /home/hpe/CDROMs/Cdrom7/*
xs      ./home/hpe/CDROMs/Cdrom7/*
```

```
# Exclude all files and subdirectories in the temporary directories.
# Handle files specified with relative and absolute pathnames
#
#   -- NOTE --  the actual directory names will still be backed up,
#               only the files within the directories will be
#               excluded.
#xs     ./usr/tmp/*
#xs     /usr/tmp/*
#xs     ./tmp/*
#xs     /tmp/*

# Don't compress files that end in ".z" or ".Z"
zs      *.[Zz]
zs *.zip
zs *.ZIP
zs *.arj
zs *.ARJ
zs *.[Aa][0-9][0-9]
zs *.[Rr][Aa][Rr]
zs *.[Ra][0-9][0-9]
zs *.[0-99]
zs *.gz
zs *.GZ
zs *.gzip
zs *.GZIP
zs *.bz2
zs *.BZ2
zs *.tgz
zs *.TGZ
zs *.tar.gz
zs *.tar.bz2
zs *.rpm
zs *.RPM
zs *.iso
zs *.ISO
zs *.mp3
zs *.MP3
zs *.asf
zs *.ASF
zs *.[Gg][Ii][Ff]
zs *.[Jj][Pp][Gg]
zs *.[Mm][Pp][Gg]
--
```

Create the file /usr/local/sbin/bru-fullbackup with the following in it. NOTE: You might want to change the label field to your tape drive and proper date

---

```
--< /usr/local/sbin/bru-fullbackup >--
#!/bin/sh
clear
```

```
# Edited 08/25/98

#HP TR4 SCSI Internal, 2.0.36, 486/160Mz/40MB, 4)IDE 3)RAID0, AHA1542 SCSI
#----------------------------------------------------------------------
#02/09/99: wrote       (3904000 KBytes), 3:28:00, 330 Kb/sec (effective)
#02/09/99: autoscan    (3904000 kbytes), 2:16:54, 475 Kb/sec

echo "Setting environment vars"
export BUFSIZE=16k
export BRUTMPDIR=/tmp
export BRUMAXWARNINGS=20000


#Only needed for old Glibc users
#export TZ=PDT

echo "Compressing old log files.  This might take a while.."
mv /var/log/bruexeclog /var/log/bruexeclog.`date +'%b%d'`
mv /var/log/bru-log /var/log/bru-log.`date +'%b%d'`
bzip2 -9f /var/log/bru-log.`date +'%b%d'`

echo "Starting BRU full backup with exclusions, compression, user intervention"
# Do not use -j, -m,
bru -c -vvvv -V -X -Z -G -L "Hp Tr4 11/27/98 - FULL" -f /dev/st0 / > /var/log/bru-log

#Only needed for old Glibc users
#export TZ=PST8PDT

# v8.8.98
#               See /etc/bruhelp for A LOT of more details
#
# Defaults to backing up "/"
#
# -c    : create (autoscan verification on by default)
#       : - if you specify  -i or -d, autoverify is disabled
#
# -d    : file comparison  (normal)
# -dd    : file comparison access mod, lengths, symlinks, ID groups
# -dddd : file comparison - hard core
#
# -e    : Estimate archive size
#
# -f    : select regular input device (same as -r)
#
# -g    : Read : Dumps the header block
# -gg   : Read : Generates ted cmd line, label, date, time, release,
#
# -h    : Print this help information
#
# -i    : inspect a archive *checksum of a directory)
```

```
#       : Not needed with "-v"
#
# -r    : Backup a raw partition
#
# -t    : List archive table of contents for files
#
# -u - use selected files
#       a - all files
#       b - block special files
#       c - character (special files)
#       d - dirs
#       l -  syms
#       p - fifos
#       r - reg
#
# -vvvv : Level 4 verbosity
#
# -w    : confirmation of each file
#
#       : wildcard expantion [must be placed in double quotes]
# -x    : restore
#
# -G    : Write a archive list (header block) at beginning of
# -L    : Label the tape
# -B    : disabled user intervention
# -D    : Enabled double buffering for faster throughput
# -Z    : compression
# -V      : execution summary w/o volume
# -X    : Exclude specific files
#
# bru -gg -f /dev/st0   : Display archive contents if written
#
#bru -vv -t -f /dev/st0 : Display entire contents of archive tape
#
#bru -x -vvvv /user/dranch/*
#
# Also, these environment variables are available in /etc/brutab
#
# Global BRU settings
#
#+OVERWRITEPROTECT=YES
#+RECYCLEDAYS=180
#+MAXWRITES=200
#+ZBUFSIZE=512k
#+SHELL=/bin/sh
#+BRUTABONLY=no
#+DEVNAMECHECK=no
#+MATCHLEVEL=2
#+MAXFILENAMELEN=255
#+READCHECKLEVEL=1
```

```
#+BRUHELP=/bru/bruhelp
#+BRUMAXWARNINGS=1000
#+BRUMAXERRORS=500
#+BRUXPAT=/etc/bruxpat
#+BRURAW=/etc/bruraw
#+BRUSMARTREST=/etc/brusmartrest
#+BRUREMOVELOG=/var/adm/bruremovelog
#+BRUTMPDIR=/tmp
--< /usr/local/sbin/bru-fullbackup End.>
```

- Ok, go ahead and insert a tape in the tape drive and run

```
"/usr/local/sbin/bru-fullbackup"
```

I usually also run "tail -f /var/log/bru-log" in another TTY to watch the progress of the backup.

- Once your backup is completed, you need to verify that you can read the files OFF the tape, restore files to different places, and also restore files back to their ORIGINAL location:

– Based on an email from the BRU mailing list:

The techniques differ depending on how the backup was created (absolute [/] or relative [.]). If you used "I" use "/" as a backup point, we are using absolute paths so (assuming you have a tape with full backups as well):

- Restore the /etc/passwd file to a different location (/tmp):

```
cd /tmp
bru  -xvf  /dev/st0  -PA  /etc/passwd
```

* the trick is "-PA" which translates absolute to relative

Now test that the files are the same:

```
diff /etc/passwd /tmp/passwd
```

- Restore the /bin/fullbru file to the same location (/bin):

```
mv  /bin/fullbru  /bin/fullbru.save
bru  -xvf  /dev/st0  /bin/fullbru
```

- Now test that the files are the same:

```
diff /bin/fullbru.save /bin/fullbru
```

- Once you are convinced that you have a good backup, now its time to create a rescue diskette.

- Download the BRU rescue diskette from:

`<ftp://ftp.estinc.com/pub/linux/Bootkit-1.01.tar.gz>`

- Here are a few other scripts that I find useful with Bru:

```
--< /usr/local/sbin/bru-viewtape >--
#!/bin/sh
clear

#echo "Starting BRU to view tape contents"
bru -gg -f /dev/st0 > /var/log/bru-tape-contents.`date +'%b%d'` 2>&1

--<end.>--
```

```
--< /usr/local/sbin/bru-find-changes >--
#!/bin/sh
clear

# Edited 01/06/99

echo "Setting environment vars"
export BUFSIZE=16k
export BRUTMPDIR=/tmp
export BRUMAXWARNINGS=20000
#export TZ=PDT

echo "Starting BRU to find all changed/missing files between tape and disk.."
bru -dd -f /dev/st0 / > /var/log/bru-diff-del-find-log.`date +'%b%d'` 2>&1

--<end.>--
```

```
--< /usr/local/sbin/bru-restore >--
#!/bin/sh
clear

# Edited 03/09/99
#
# NOTE:  This script is run as:  "/usr/local/sbin/bru-restore /home/username"
#          where the "/home/username" is the path and/or the full path and filename
#          of the data you want to restore.  Bru will then find this data on the
#          tape and restore it to its original location.  If you want to restore
#          the file to a DIFFERENT location, please consult the manual for
#          "absolute to relative path translation"
#

echo "Setting environment vars"
export BUFSIZE=16k
export BRUTMPDIR=/tmp
export BRUMAXWARNINGS=20000
#export TZ=PDT

echo "Compressing old log files.  This might take a while.."
mv /var/log/bru-restore-log /var/log/bru-restore-log.`date +'%b%d'`
mv /var/log/bruexeclog /var/log/bruexeclog.`date +'%b%d'`
```

```
bzip2 -9f /var/log/bru-restore-log.‘date +’%b%d’‘

echo "Starting BRU partial restore "
# Do not use -j, -m,
bru -x -vvvv -f /dev/st0 $1 > /var/log/bru-restore-log
--<end.>--
```

29.4 Using a CD-R or CD-R/W drive

See 39 (Section 39) for full details.

30 SSH Terminal, FTP, X-windows, and tunnel encryption

30.1 What is SSH and the differences between SSH protocol v1 and v2

SSH is both a protocol and a program suite that allows for TELNET-like CLI communications, FTP, and the ability to create VPN connections while having all of it ENCRYPTED. For me, I always use SSH because if I was to login with non-encrypted programs like TELNET, FTP, POP-3, etc., all of my username/passwords (and all following traffic) would go over the Internet in CLEAR-TEXT. * THIS IS BAD! * What's even cooler is you can actually use SSH to encrypt NON-secure systtems like TELNET and POP3 if need be.

So why is non-encryted communications bad? For example, say some evil person was between your local machine and your POP-3 server. If they were to sniffing the traffic, not only would they be able to get your username / password but also get all of your transmitted email too! Now you might be thinking this is paranoid thinking but securing your connections isn't hard and you should be better safe than sorry.

Using SSH, ALL traffic is encrypted. Plus.. it can actually ease the setup of remote Xwindows connection and even speed things up with the use of built-in SSH compression!

NOTE: SSH comes in two flavors and two versions. SSH protocol Version 1 and Version 2 from both OpenSSH and SSH.com

- v1: (no longer recommended) SSH Version 1 is much better than simply using clear-text TELNET and it is supported by other many other Unix and Windows clients. It also supports fast ciphers (Blowfish and IDEA) and file transfer (scp). The major benefit of SSHv1 is that it is completely free for both end users and commercial companies. In recent times, tools have become available that can decrypt SSHv1 traffic on the fly thus removing most security from encrypting the traffic with SSHv1.

  Like SSHv1, SSHv2 supports fully encrypted communications but also supports encrypted FTP file transfers (sftp) in addition to the original scp.

  So why is there a version 2 other than just adding sftp functionality? There are some fundamental flaws in the SSHv1's protocol. Since the protocol itself was flawed, SSHv1 was discontinued. People complained that v1 could be fixed and the licensing of v2 was too restrictive (and expensive from SSH.com). Fortunately, SSH.com (the original authors of SSH) somewhat relaxed their licensing for SSHv1 and SSHv2 for both Linux and *BSD. Unfortunately, SSHv1 has been deprecated for some time, had some serious recent security issues, and ultimately is no longer supported by SSH.com. Due to the lack of modern support (patches, etc.) and support for SSHv2 clients is very common, I do NOT recommend users to install SSHv1 or run SSHv1 compatibility mores anymore.

- v2 Version 2 is a re-worked and stronger version of SSH. In addition to all the functionality in SSHv1, version 2 brings encrypted FTP, support for digital certificates and PKI, and many other features.

> Unfortunately, SSHv2 does not support the fast Blowfish or IDEA ciphers but the other ciphers aren't much slower now (AES, etc.).
>
> Unfortunately, most SSH v1 clients (like SecureCRT v2.x for Windows) -CANNOT- connect to a v2 server unless the server is compiled up to support "compatibility" mode. Please note that SecureCRT v3.x now supports both SSHv1 and SSHv2. I recommend to upgrade your SSHv1-only clients to support SSHv2 rather than support the deprecated SSHv1 protocol.

I used to recommend the use of the SSHv2 service along with SSHv1 compatibity mode but I can't recommend this any longer. With SSHv1 being no longer supported, the recent CRC32 Compensation Attack vunerability, and the fact that there are enough good comercial/free SSHv2 clients out there, we can finally get rid of SSHv1 servers and clients. But, if this doesn't work for you, just be sure to keep up Bugtraq for any known SSHv1 exploits, etc.

NOTE: I have personally noticed that when connecting to SSHv2 servers running in SSHv1 Compatibility mode, the initial connection time until you receive a prompt is SIGNIFICANTLY slower than SSH v1 servers. Oh well.

NOTE #2: The following example does showshow to install both SSHv1 and SSHv2 to support both types of connections. If you don't want to run SSHv1 (because it's old) or SSHv2 (because of licensing issues), simply skip that section.

30.2 Running OpenSSH vs. SSH.com code

So you might be asking yourself, why is there both a commercial and free version of SSH? Well, the people at SSH.com orginally created SSHv1 and later, SSHv2. Understandably, they needed to make money from their work it so they charged ALL users for the use of it. This annoyed many people from the OpenBSD camp and thus they started to write their own version of SSH that would always be free. Over the years, SSH.com changed their licensing where it was now free to use for NON-commercial use for the Linux and *BSD operating systems. If used in a commercial setting or you wanted to run it on Solaris, HPUX, AIX, etc., it was still quite pricey.

Another reason why OpenSSH came to be was that SSH.com wanted to open up the the SSH protocol to become a standard. For this to happen, the various standards bodies required that the protocol be implimented by at least one 3rd party. Fortunately for SSH.com, the OpenSSH and OpenSSL people were already working on it.

So which do I recommend to you? Well, first, I recommend you review what SSH.com considers "NON-Commercial" use. JUst bring up a web browser and look at their LICENSING terms (they are surprisingly readible). After reading that, if you have no money and work in a a commercial environment, you probably need to run OpenSSH. Even if you work in a non-commercial environment, they have the right to change their minds again. As linux becomes more and more popular, you can plan on it to some extent. Ultimately, that would be a support nightmare going from SSH.com to OpenSSH. If your're starting fresh, why not just start with OpenSSH?

The main reasons why you might want to go with SSH.com's code are things like:

- online or telephone support
- robust digital certificates and PKI support

30.3 OpenSSH: Thoughts, Issues, and Features

OpenSSH uses OpenSSL for it's encryption libraries. Because of this, you need to install OpenSSL before you install OpenSSH. Currently, this is not covered in this section but should be easily added via a RPM, PKG, DEB, or the "use the source Luke!". If enough people ask for it, I can add OpenSSL instructions to TrinityOS. Anyway, you should verify that the version of OpenSSL on your machine is v0.9.5a or newer due to security issues. To do this, run the command:

```
  openssl version
```

For users that still use SSHv1, OpenSSL 0.9.5a+will not properly support Blowfish over SSHv1 connections. This shouldn't be an issue as the use of SSHv1 is NOT recommended. You should strive to ONLY use SSHv2 in your environment.

Features:

Before you install OpenSSH, you should know something about OpenSSH 3.x. OpenSSH has a powerful chroot mechanism called "Privilege Separation". With this system in place, even an exploit against OpenSSH should only get user-level access and NOT root access. This system now mostly works on all systems now but there are a few corner cases. Specifically, some Linux kernels make this feature incompatible with SSH compression. If you use compression (I do), I recommend to avoid the use of this feature for now.

If you do want to use Privilege Separation, you need to setup the CHROOT environment *FIRST*:

```
        mkdir /var/empty
        chown root:sys /var/empty
        chmod 755 /var/empty
        groupadd sshd
        useradd -g sshd -c 'sshd privsep' -d /var/empty -s /bin/false
```

30.4 Compiling OpenSSH:

- Goto your local OpenSSH mirror shown in 5 (Section 5) and download the newest code. This archive includes both the server and client code.
- Uncompress the OpenSSH archive and configure it up:

```
    ./configure --prefix=/usr/local --sysconfdir=/etc/ssh --with-pam \
    --with-tcp-wrappers --with-md5-passwords --with-ipv4-default
```

- Compile it and install it with:

```
    #Creates the binaries
    #
    make



    #Installs the code to the following places:
    #
    #  Configs: /etc/ssh    (might conflict with ssh.com installations)
    #
    #  Client:  /usr/local/bin
    #  Server:  /usr/local/sbin
    #
    make install
```

- If that went well, the system should have created a new system key pair. Just to check, try running the following. It won't do anything if the system did infact create the new keys.

```
make host-key
```

- Finally, make sure the SUID Root bit is removed from the SSH daemon (server). This will insure that non-root users cannot use privleged ports (1-1023). This also can also help increase system security if some exploit comes out against OpenSSH.

```
#Remove the SUID root-bit
#
chmod 755 /usr/local/sbin/sshd
```

30.5 Compiling up SSH.com's SSH

- Go to the SSH archive shown in 5 (Section 5) and download the newest version of the v2 SSH server (and optionally SSHv1 code) - these archives also include the SSH client as well.
- Un-tar the v2 SSH server/client (optionally the SSHv1) archives by running:

```
tar -xzvf ssh-3.2.x.tar.gz

#OPTIONAL - and not Recommended
#
#   If you plan on installing SSHv1, decompress that archive now as well
#
tar -xzvf ssh-1.2.x.tar.gz
```

- OPTIONAL: If you still want to install SSHv1 support, do the following:

  NOTE: If you want to support both SSHv1 and v2 clients, you MUST install SSH v1 first. To do so, "cd" into the SSHv1 source code directory and:

  -

```
                #For SSHv1 only
                                ./configure --with-libwrap --disable-suid-ssh
```

    This tells SSH to set itself up for this particular hardware setup with:

    * support TCP wrappers as configured by /etc/hosts.*
    * to NOT install itself as SUID root
    * run multiple copies of GCC for better compile times

```
                                "make -j4 clean"
                                "make -j4 "
                                "make install"
```

  For SSH v2 server support (using /etc/hosts.allow without IPv6 support and without built-in SSHv1 compatibility support):

```
cd ssh-3.2.x

./configure --with-libwrap --disable-suid-ssh --without-ipv6 \
 --without-internal-ssh1-compat
```

This tells SSH to set itself up for this particular hardware setup with:

- support TCP wrappers as configured in /etc/hosts.*
- to NOT install itself as SUID root
- run multiple copies of GCC for better compile times

```
"./make  -j4 clean"
"./make -j4"
"./make install"
```

NOTE: The "make install" command might take some time (key generation does 7 passes) and time per pass depends on your Linux box's CPU power.

30.6 Configuring OpenSSH or SSH.com to load the server daemon upon reboot with startup scripts

Next, you need to have the SSH daemon load upon every reboot

Basically, there are two ways to do it. One is the Sys-V way (Redhat, Solaris, etc) or the BSD way (Slackware, SuSe, etc). Please see the middle portition of 8 (Section 8) to see if you had disabled SSHd from starting upon reboot.

NOTE: When loading the SSH daemon, lower the "xx" numbers Sxx.sshd or eariler in the rc.local, the faster the box will come back up with SSH support after a reboot.

For me with a CD-ROM changer, if the SSHd daemon was after the rc.cdrom startup script file, I would have to wait until all 7 CD-ROMs were mounted before SSHd begins to load! A slow process indeed!

For SysV machines (Redhat, etc):

/etc/rc.d/init.d/sshd

```
--
#!/bin/bash
#
#       /etc/rc.d/init.d/sshd
#   v1.2
#
# sshd          Start the Secure Shell daemon
#
# chkconfig: 345 12 12
# description: The Secure Shell daemon, versions 1 and 2, allows for strong \
#              authentication, encrypted communications and tunnels with \
#              remote clients also using SSH.
# processname: sshd
# pidfile: /var/run/sshd.pid
# config: /etc/sshd_config

# v1.2 - Support for OpenSSH (default setting) added
# v1.1 - Fixed an error where it was starting SSHD and not SSHD2
# v1.0 - initial release
```

```
#
# Source function library.
. /etc/rc.d/init.d/functions

# OpenSSH settings - Add #s in front of the following lines if you want to
#    use SSH.com code
#
# (enabled by default)
#
SSHD=/usr/local/sbin/sshd
SSHD_CONFIG=/etc/ssh/sshd_config


# Disabled ssh.com settings - remove the #s if you want to use SSH.com
#
# (disabled by default)
#
# SSHD=/usr/local/sbin/sshd2
# SSHD_CONFIG=/etc/ssh2/sshd2_config

# If you are running SSHv1 in addition to SSHv2, uncommend out the
# following lines
#
#SSHD1=/usr/local/sbin/sshd
#SSHD1_CONFIG=/etc/sshd_config

case "$1" in
    start)
        echo -n "Starting SSH services: "

        if [ -x $SSHD -a -f $SSHD_CONFIG ]

    # If also running SSHv1, # out the line above and un-# the line below
    #if [ -x $SSHD1 -a -f $SSHD1_CONFIG -a -x $SSHD -a -f $SSHD_CONFIG ]

        then
                daemon $SSHD
        else
                echo_failure
        fi
        echo
        touch /var/lock/subsys/sshd
        ;;
    stop)
        echo -n "Shutting down the SSHd daemon: "
        killproc sshd
        echo
        rm -f /var/lock/subsys/sshd
        ;;
    status)
```

```
        status sshd
        ;;
    restart)
        $0 stop; $0 start
        ;;
    reload)
        killall -HUP sshd
        ;;
    *)
        echo "Usage: sshd {start|stop|status|reload|restart}"
        exit 1
        ;;
esac
```

To activate this new script, run the following command:

```
        chkconfig --level 345 sshd on
```

For BSD-style machines (Slackware, etc): ————————————————-

Edit the following file and put the text toward the TOP of the file:

/etc/rc.d/rc.local

```
--
echo "Starting sshd v2 with Compatibility mode..."
/usr/local/sbin/sshd
--
```

30.7 Configuring the Unix services

Most machines should have this first step already done but just make sure it's there:

Edit "/etc/services", find where port "22" should go and add this line (if it isn't there already):

```
                ssh             22/tcp
```

30.7.1 Configuring OpenSSH:

Ok, time to configure SSH:

- Configure the SERVER by editing the /etc/ssh/sshd_config file and add/change the following. You can find more info by reading up on the "sshd_config" man page.

```
     #Disable the use of SSHv1 on this server (remove the "1")
     Protocol 2

     #Disable the ability to log in as root
     PermitRootLogin no

     #Make sure all accounts have to have passwords
```

```
PermitEmptyPasswords no

#Allow X Forwarding
X11Forwarding yes

#Disable this for hosed reverse DNS
#VerifyReverseMapping no

#Disable Privilege Separation - required if you plan to use compression
#                                with OpenSSH v3.x on certain OSes
UsePrivilegeSeparation no

#Enable compression by default - Privilege Separation must be disabled
Compression yes
```

- Next, configure the client by editing the /etc/ssh/ssh_config file and add/change the following. You can find more info by reading up on the "ssh_config" man page.

```
#Allow to forward X over SSH
ForwardX11 yes

#For hosed reverse dns
#CheckHostIP no
```

30.8 Configuring SSH.com SSH:

***** If you installed SSH.com SSH v2 but STILL want to support SSH v1 clients (not recommended), etc., do the following:

- edit /etc/ssh2/sshd2_config and either verify or add the following lines to the section that is under "*:". If any of the following lines do exist but have a "#" in front of it, delete the "#" and edit the line to look as follows:

```
/etc/ssh2/sshd2_config
--
Ssh1Compatibility   yes
Sshd1Path           /usr/local/sbin/sshd1
--
```

- It should also noted that if you are concerned with absolute security and don't need the following function, I recommend to do the following:

```
/etc/ssh2/sshd2_config
--
#If you don't need SSH tunnels, disable them by putting a "#"
#in front of the line:
ForwardAgent            yes

#If you don't need X11 SSH forwarding, disable it by putting
# a "#" in front of the line:
ForwardX11          yes
--
```

- I also recommend to disable the ability to login via SSH1/2 as root. To do this, edit the following files and change them to read:

```
/etc/ssh2/ssh2d_config
--
PermitRootLogin no
--
```

- Next, edit

```
/etc/sshd_config
--
PermitRootLogin no
--
```

- Next, edit /etc/ssh2/ssh2_config and either verify or add the following lines to the "*:" section. If the line does exist but there is a "#" in front of it, delete the "#"

```
/etc/ssh2/ssh2_config
--
Ssh1Compatibility   yes
Ssh1Path           /usr/local/bin/ssh1
--
```

30.9 Configuring BASH aliases for proper SSH operation through firewalls

- Next, I would recommend to add the following line towards the bottom of /etc/bashrc:

```
alias ssh='/usr/local/bin/ssh -C -P -c blowfish'
alias scp='/usr/local/bin/scp -C -c blowfish -L'
```

What this does is when you SSH out of the Linux box itself, SSH will:

- Use Compression if possible
- If compression is enabled, use the Blowfish compression codec. Why Blowfish codec vs. say 3DES? Because its FASTER.
- Disable the R-tools emulation of using ports < 1024 (this is the -P and -L options)

Please note that for this alias to take effect, you will have to log out and then re-login.

- Now you need to either load or RE-load the SSH server.

30.10 Starting the SSH server:

If you don't currently have a SSHd server running, simply type in the following to test it out:

```
/usr/local/sbin/sshd
```

Hopefully, you will just get the command prompt back and the SSH server will be running in the background.

If you already have a SSH v1 server running, things get a little more complicated:

- You need to either login to the console of the Linux server or TELNET (yes.. TELNET and not SSH) into your Linux box. Also, if you are going to TELNET in and you are running a strong firewall rule set, you'll have to allow TELNET into your firewall.
- Now, login to your box WITHOUT SSH and kill the running SSHd process:

```
#SYS-V style (redhat):
#
         killall -HUP sshd

#BSD-style (Slackware):
#
         kill -HUP ‘ps aux | grep sshd | grep -v -e grep | awk ’{print $2}’‘
```

- Finally, start the SSHd process

```
/usr/local/sbin/sshd
```

That's it! The SSH server should be running now! If there seems be be problems or the server doesn't load, see below for some troubleshooting ideas. If things DO seem to be running, load up your SSH client and try it out. To SSH from your Linux box, just run "ssh username@xyz" where the "username@" can be left blank if you want to use the current username you're already logged in as or a different username and udquot;xyz" is the remote SSH-enabled server's fully qualified domain name or IP address.

30.11 SSH Problems? Here are a few possible solutions

1. Are you getting the error "WARNING: Privilege separation user "sshd" does not existd" from OpenSSH? If so, you either forgot to create the SSHd user as shown above or you didn't disable priviledge separation in the /etc/sshd_confif file (disabled by default in TrinityOS)
2. Can't connect to your SSH server from a host out on the Internet? Make sure that if you are using a IPTABLES / IPCHAINS / IPFWADM firewall, that port 22 is allowed IN and OUT.
3. Does SSH initially make a connection and then disconnect? Make sure that if you are using TCP Wrappers, /etc/hosts.allow, that SSH access is allowed in from the requesting remote machine's FQDN or IP address.
4. If you can SSH out from a MASQed PC but NOT from the Linux server itself AND you are getting firewall hits in /var/log/messages that look something like:

```
  Jul  6 10:28:49 roadrunner kernel: Packet log: output REJECT eth0 PROTO=6
100.200.300.19:716 212.222.333.222:22 L=60 S=0x00 I=5107 F=0x0000 T=64 SYN (#38)
```

   What is happening is that you didn't follow the above requirement to add an SSH alias to your /etc/profile and have SSH run with the "-P" option. Specifically, the SSH packet leaving the server is using LOW ports (in this example, port 716).

30.12 SSH Port Forwarding

FULL SSH port forwarding!

UNIX access:

SSH PORTFWDing is a method to tunnel or "VPN" traffic through an SSH server. So not only can you transparently gain access to remote systems, you can tunnel non-encrpted applications like TELNET, FTP, etc. through an encrypted SSH connection. Here is how you can configure a SSH client for secure IMAP, SNMP, and LDAP access through a SSH tunnel. Also know that other people can setup these tunnels to YOUR SSH server if they have the proper access.

NOTE: One VERY cool thing about this setup is that the server that has the SSH server does NOT have to be the server you need to access. What this means is that the SSH server can actually terminate the tunnel on the edge of the remote network but then FORWARD the PORTFW traffic to a specific intended INTERNAL server. Very cool.

To setup this tunnel, I recommend to create a script called "start-tunnel". This script assumes that "some.remote-ssh-server.com" is the SSH server and that "some.internal-mail-server.com" is the internal server that you ultimately want to connect to (for this example, that internal machine is a mail server).

start-tunnel

```
echo Forward IMAP, LDAP, SMTP to allegro
/usr/local/bin/ssh.old -C -P johnjoe@some.remote-ssh-server.com \
        -L 143:some.internal-mail-server.com:143 \
        -L 25:some.internal-mail-server.com:25 \
        -L 389:some.internal-mail-server.com:389 sleep 7200
```

Lets break this script out to better understand it:

```
        1) this example uses the older SSHv1 client.  If you get an
           error like:

             "Executing /usr/local/bin/ssh1 for ssh1 compatibility.
                Bad forwarding specification '143'."

                   This means that the remote SSH server is NOT supporting
                   SSHv2.  So, this is why I hard coded it to use SSHv1.

                        2) -C means use compression
                        3) -P means to NOT use ports less that 1024 (privileged ports)
                        4) "johndoe" is the login on the remote SSH server
                        5) "some-remote-ssh-server.com" is the remote SSH server

                        6) "-L 143 some.internal-mail-server.com:143" means:
                                A) I want to forward all LOCALHOST traffic to port 143
                                B) Send this traffic to "some.internal-mail-server.com"
                   on port 143

                                NOTE:  If you didn't catch that, it will be forwarding
                               ******  your LOCALHOST traffic on port 143 to that remote server.
                                         SO, if you were originally configuring your IMAP client
                                         to directly connect to "some.internal-mail-server.com",
```

```
                    you will now have to re-configure it to connect to "localhost".
                    Weird huh?  Once the SSH tunnel comes up, it will work
                    completely transparently.

        One trick several people like is to create an /etc/hosts.ssh
        file.  In this file, add the line:

             some.internal-mail-server.com     127.0.0.1

        With this in place, add some lines to your SSH PORTFW
        script that will rename your original /etc/hosts file and
        use this /etc/hosts.ssh file in it's place.  When this
        happens and your email client comes up, it will check the
        /etc/hosts file FIRST before going to DNS.  So, when SSH
        PORTFWDing is running, your email client will automatically
        use the PORTFW connection.  If SSH is down, it will use DNS.
        Plain and sweet huh?

            7) Repeate the forwards for SMTP and LDAP as well

            8) Like RSH, SSH will execute the command "sleep 7200"
   on the remote server.  So, after 7200 seconds or 2 hours, the tunnel
   will shut down.
```

* For other UNIX examples, please see the SSH section in 5 (Section 5):

Windows access:

- If you are looking for a great SSH client for Windows, check out SecureCRT at `<http://www.vandyke.com>`. Here is an example how to setup SecureCRT perfectly for Linux.

——— NOTE: This SCRT configuration example shows how to configure SecureCRT to both enable SSH encrypted communications to the remote host but also enable transparent SSH port forwarding for ALL POP-3 and communications to that same given server. If you also want to encrypt additional protocols like IMAP4, etc., just use this configuration as this as a template.

Please note that to enable SSH port forwarding, a normal SecureCRT SSH connection needs to be established FIRST to your remote server. Once the SSH connection is running, all POP-3, etc communications will be transparently encrypted! You won't even notice its doing it.

Once the SSH connection is down, all POP-3, etc communications will break because the given POP-3, etc clients must be reconfigured to connect to IP address 127.0.0.1. More on this in a moment.

———

- File –> Quick Connect –> "Session list" tab –> New
- Enter in the name of a SSH site to connect to
- Change the protocol to "SSH"
- Enter in the fully qualified domain name of the remote site
- Verify the port is set to "22"
- Enter in your username for the remote site

- Change the Cipher to "blowfish"
- Change the authentication to "password"

I would also recommend to do the following:

Session–>Advanced–>

General tab:

- Enable "Use Compression" at a level of 5

Port Forwarding: - Click on the NEW button

- Local port: 110
- Remote Hostname: roadrunner.acme123.com
- Remote port: 110
- Save
- Enable "Forward X11 packets"
- Save

Emulation

- vt102 and enable ANSI color
- Change the Scollback buffer to "9999"

Options

- DISABLE "Scroll to bottom on output"

- You have to do one last thing for SSH forwarded connections. You need to reconfigure your POP-3 client, say Netscape or Eudora, to connect to 127.0.0.1 and -NOT- your normal POP-3 server. What this does is the POP-3 client will conenct to 127.0.0.1 (localhost on your local machine) and then SecureCRT will SSH it and forward it over the first configured instance of SCRT with port 110 forwarded. As mentioned above, you can create a batch file that swaps around the C:\WINDOWS\HOSTS file and let you not have to reconfigure your applications. See above in the Unix PORTFWD section for more details.

NOTE: If you have multiple POP-3 clients running, this will be a problem since you can't port forward port 110 twice. To fix this, you will have change the POP-3 client to use a different port other than port 110 (say port 123) and then configure that SCRT sesstion profile to SSH forward port 123 to remote port 110. Get it?

NOTE2: SSH port forwarding does NOT work well with ACTIVE-style ftp connections. Re-configure your FTP clients to use PASV connections on port 21 and then SSH'ed FTPs will work ok.

———–

- That's it. From S-CRT, go ahead and try connecting to your remote SSH server and you should be prompted with a dialog box asking to "Accept and save" the keypair. Click on "OK". Now you should be prompted to enter in your password and you should now login over an SSH encrypted connection! With the SSH connection running, now all your POP-3 traffic will also be transparently encrypted to make your username/password and files safe from prying eyes.

31 Software RAID 0 (striping) Hard drives

If you didn't notice in 4 (Section 4), this TrinityOS enabled server (Roadrunner) has (7) hard drives and (2) CD-ROMS running on it now. Four IDE HDs are in the main system case and the other (3) SCSI HDs and (1) tape drive is in an old AT-style computer case.

To pull this off, I ordered a SCSI cable that has (2)external HD50pin SCSI-2-Fast connectors on it and 8 internal SCSI 50-pin internal ribbon cable connectors in the middle. I bought this from `<http://www.corpsys.com>` [part num: SCSI28] for ~$59. I then used one of my old AT-style cases with its power supply. With all this, I now have a external RAID box! It's no hot-swap cage but it works. Anyway, the following section will tell you how to implement RAID 0 (Striping) in software. Changing the configs to Linear, RAID-1, or RAID-5 won't be hard as long as you can afford the lost capacity or afford the extra disks.

- Download ALL the various version of the RaidTools from the URL in 5 (Section 5)

The reason to download ALL of the available versions is that I've noticed that some of the versions in the past would NOT compile. Other versions didn't have all the docs, etc. In the past, the Raidtools has been in in a sad state right now but it DOES work nicely once you put it all together.

NOTE:

You will notice that there is both a Software-RAID HOWTO and a Software-RAID-0.4x on the various Linux mirrors. The reason for this is that the 0.4x HOWTO only covered the 2.0.x kernels and was more of a FAQ. The new howto covers the newer 2.2.x Software RAID (via a patch) or the 2.4.x kernels.

Anyway, from here on out, assume I'm using the new Raidtools-0.90 system

- Download and install the newest available kernel found in Section 5 into /usr/src/kernel/linux

- Next, download the newest Raidtools patch for your kernel (URL is in section 5 and also put it in /usr/src/kernel/linux. Don't worry about this code being in the "Alpha" directory, this stuff is VERY stable.

- Apply the patch by running the following comment (for a 2.2.19 kernel): patch -p1 < raid-2.2.19-A1

- Now run "make config" (if you haven't already done this as shown in 11 (Section 11))

- Configure the kernel as you normally would but, in the HD hardware support section, enable the following (you can make these modules if you wish but I recommend the monolithic approach):

```
Multiple devices driver support (CONFIG_BLK_DEV_MD) [Y/n/?] Y
Autodetect RAID partitions (CONFIG_AUTODETECT_RAID) [Y/n/?] Y
   Linear (append) mode (CONFIG_MD_LINEAR) [N/y/m/?] N
   RAID-0 (striping) mode (CONFIG_MD_STRIPED) [Y/m/n/?] Y
   RAID-1 (mirroring) mode (CONFIG_MD_MIRRORING) [Y/m/n/?] Y
   RAID-4/RAID-5 mode (CONFIG_MD_RAID5) [Y/m/n/?] Y
   Translucent mode (CONFIG_MD_TRANSLUCENT) [Y/m/n/?] N
   Hierarchical Storage Management support (CONFIG_MD_HSM) [N/y/m/?] N
      Boot support (linear, striped) (CONFIG_MD_BOOT) [Y/n/?] Y
```

- Now make the kernel as normal with either "make dep; make clean; make bzImage; make modules; make modules_install" or just use TrinityOS's "built-it" script.

- Now, install the kernel into lilo, LOADLIN, etc. and reboot (shown in 13 (Section 13) & [Section 14]).

- Once the box has rebooted, you might not need to compile up the Raidtools-0.90 archive. To verify this, try running "/sbin/mkraid -V". If the program is found and it reports version 0.90.0 then you don't need to do anything. If the program is NOT found, please follow these instructions:

- Uncompress the raidtools-0.90 archive ("tar -xzvf" for .tar.gz or "tar xvIf" for tar.bz2)

- cd into the created directory and run "./configure"

- Then run run "make all" and "make install"

- Hopefully everything went ok

- Now that you have the utilities and your kernel is ready to do, you need to edit your system init files to properly bring up the md0 software-raid service.

!!!NOTE!!! These example configs ASSUME that the partitions to be RAIDed are /dev/hda1 and /dev/sda1. Modify your configs to reflect your own environment!!!

!!!NOTE #2 Some distributions support Software-RAID automatically. To verify if this is so, look in the /etc/rc.d directory with this command:

"rgrep -r -i raid /etc/rc.d"

If anything is found (Redhat and Mandrake have it configured in /etc/rc.d/rc.sysinit), you can just use that setup though they are out of date with the use of "Auto-Dectection" partitions.

- To create a "Auto-Detected" RAID partition, you need to set each one of the HD's RAID partition to type "fd" and NOT the normal ext2, reiserfs, etc.

```
# /sbin/fdisk /dev/hda

The number of cylinders for this disk is set to 1860.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 1860 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1             1      1860  14940418+  fd  Linux raid autodetect

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
```

- For users that don't want to use Auto-Detect RAID or those users without a RAID-enabled distro, create the following file:

/etc/rc.d/rc.raid

```
#!/bin/sh

# See how we were called.
case "$1" in

    start)
       #Start up the RAID subsystem - not needed for auto-detect
       /sbin/mkraid /dev/md0
       echo "Disks added"
       /sbin/raidstart /dev/md0
       echo "Raid -RAID0- started on /dev/md0"
    ;;

    manual)
       #Start up the RAID subsystem - not needed for auto-detect
       /sbin/mkraid /dev/md0
       echo "Disks added to /dev/md0"
       /sbin/raidstart /dev/md0
       echo "Raid RAID0 started on /dev/md0"
       /bin/mount -t ext2 /dev/md0 /mnt/raid
    ;;

    stop)
       echo "/dev/md0 umounted"
       /bin/umount /dev/md0
       echo "/dev/md0 stopped"
       /sbin/raidstop /dev/md0
    ;;

    *)
       echo "Usage: rc.raid {start|stop}"
       exit 1

esac
exit 0
```

Once you have created this script file, make it executable by running "chmod 700 rc.raid"

+++ Older Redhat users (5.0-5.2), edit the /etc/rc.d/rc.sysinit (find the following lines and insert the following lines (around line 159):

```
/etc/rc.d/rc.sysinit
--
if [ -x /sbin/kerneld -a -n "$USEMODULES" ]; then
    if [ -f /proc/sys/kernel/modprobe ]; then
        # /proc/sys/kernel/modprobe indicates built-in kmod instead
        echo "/sbin/modprobe" > /proc/sys/kernel/modprobe
    else
        /sbin/kerneld
        KERNELD=yes
    fi
```

```
fi

# Start the initialization of the MD0 RAID service
/etc/rc.d/rc.raid start

# Check filesystems
if [ ! -f /fastboot ]; then
echo "Checking filesystems."
fsck -R -A -V -a $fsckoptions
.
.
.
--
```

+++ Slackware users, edit the /etc/rc.d/rc.S file, find the following text and append the following:

```
/etc/rc.d/rc.S
--
# remove /etc/mtab* so that mount will create it with a root entry
/bin/rm -f /etc/mtab* /etc/nologin /var/run/utmp \
  /etc/shutdownpid /var/run/*.pid

# Start the initialization of the MD0 RAID service
/etc/rc.d/rc.raid start
--
```

All Distributions:

Though I recommend to read the Software-RAID HOWTO to get all the details, here is an example for:

- A RAID-0 (striped or additive capacity) RAID setup - with (2) HDs on - /dev/hda1 - /dev/sda1

/etc/raidtab

```
raiddev /dev/md0

#Linear is "linear", RAID0-stripe = "0", RAID1-mirror = "1", RAID5-volume = "5"
raid-level      0

#Number of drives you are RAIDing together
nr-raid-disks   2

#File system stuff
persistent-superblock 1

#Changing this will change performance for your system based on file
# sizes, placement, etc.  Dont change this unless you plan to reformat
# the RAID volume.
chunk-size     4

#List and number the drives in the RAID volume
device          /dev/hda1
```

```
raid-disk       0
device          /dev/sda1
raid-disk       1
```

NOTE: There is several raidtab options that can increase performance, etc (stripe size, Inodes..). For now.. I'm just shooting for functionality but the stock performance is pretty good. Please see the Software-RAID howto for more details.

- Ok, so lets start things up MANUALLY to make sure things are ok.

- FIRST, triple check the /etc/raidtab file!! If you have the wrong drive or partition in there, KISS THAT DATA GOODBYE!

- Ok, run the command "/sbin/mkraid /dev/md0". You should see something like the following:

```
handling MD device /dev/md0
analyzing super-block
disk 0: /dev/hda1, 14940418kB, raid superblock at 14940352kB
disk 1: /dev/sdb1, 8890352kB, raid superblock at 8890240kB
```

- Next, make sure the kernel thinks things are ok

```
# cat /proc/mdstat

Personalities : [raid0] [raid1] [raid5] [translucent]
read_ahead 1024 sectors
md0 : active raid0 sdb1[1] hda1[0] 23830592 blocks 4k chunks
unused devices: >none<
```

- Ok, if all is well, just format the /dev/md0 device with your filesystem of choice. For me, I still use EXT2. So, as an example, just run:

mke2fs /dev/md0

NOTE: There is some mke2fs options to increase performance, etc (stripe size, Inodes..). For now.. I'm just shooting for functionality but the stock performance is pretty good. Please see the mke2fs man page for details.

- Once things are formatted, mount it:

mkdir /mnt/raid mount /dev/md0 /mnt/raid

if things went ok, you should have just received the UNIX prompt. So.. check it with the "df" command:

```
               # df
               Filesystem       1k-blocks      Used Available Use% Mounted on
               /dev/sda7          2055600   1470712    480468  75% /
               /dev/md0          23456268        20  22264720   0% /mnt/raid
```

- Ok, so lets make sure this is mounted after reboots, etc. edit the /etc/fstab file to automatically mount this new RAID setup to some mount point. Please note that TrinityOS does NOT cover booting root partitions (/) off of Software-RAID setups. Please see the Software-RAID howto on how to do this.

Anyway, here is an example of mounting the RAID setup on /mnt/raid:

```
#RAID volume    mount point     FileSys  FS options   Dump fsck order
/dev/md0        /mnt/raid0       ext2    defaults        1 2
```

- For older setups or people NOT using Auto-Detect RAID:

- Go ahead and type in "/etc/rc.d/rc.raid start"

- If you get any errors about /dev/md0 not existing, run the command "/dev/MAKEDEV md0" and the run the script again. Yes.. use the CAPs.

- Ok, things are cool! Reboot! Make sure things are STILL cool!

32 SCSI CD-ROM Changers: Installing and Setup

Most SCSI CD Changers use one SCSI ID number and then use LUNs (Logical Unit IDs) to address each CD within the changer. With LUNs, now you can access all 4-?12? CDs in the changer from a single SCSI ID. Problem is, not all changer's LUN systems work with Linux.

Because of this, you will have to experiment with the kernel option for Multi-LUN scan support. With my Nakamichi 7-CD changer (old 2x-speed), if I enable the multi-LUN support, my kernel would HANG after the box would post the SCSI changer device but before it was to post an additional single CD CD-ROM drive. By turning OFF the Multi-LUN kernel option and recompiling, my box would boot fine.

So, with that in mind:

- Try to NOT ENABLE the:

Probe all LUNs on each SCSI device (CONFIG_SCSI_MULTI_LUN) [N/y/?]

option unless your changer is NOT properly recognized.

- Add the changer to the SCSI chain and boot up the linux box.

- Create the following file: /etc/rc.d/rc.cdrom

NOTE: Please note that the UID and GIDs are specific to my machine and you will need to change them for your system. UIDs are defined in /etc/passwd and GIDs are defined in /etc/groups.

NOTE2: The permissions of these CDROMs after mounting STILL isn't right. I'm working on it but I have to admit I'm stumped.

/etc/rc.d/rc.cdrom

```
--
#!/bin/sh

# See how we were called.
case "$1" in
 start)

  echo "Mounting CD-ROMs.."

  mount -t iso9660 /dev/scd0 hpe/CDROMs/Cdrom0 -o norock,uid=501,gid=10,suid,mode=0550
  mount -t iso9660 /dev/scd1 hpe/CDROMs/Cdrom1 -o norock,uid=501,gid=10,suid,mode=0550
  mount -t iso9660 /dev/scd2 hpe/CDROMs/Cdrom2 -o norock,uid=501,gid=10,suid,mode=0550
  mount -t iso9660 /dev/scd3 hpe/CDROMs/Cdrom3 -o norock,uid=501,gid=10,suid,mode=0550
  mount -t iso9660 /dev/scd4 hpe/CDROMs/Cdrom4 -o norock,uid=501,gid=10,suid,mode=0550
```

```
 mount -t iso9660 /dev/scd5 hpe/CDROMs/Cdrom5 -o norock,uid=501,gid=10,suid,mode=0550
 mount -t iso9660 /dev/scd6 hpe/CDROMs/Cdrom6 -o norock,uid=501,gid=10,suid,mode=0550
#  mount -t iso9660 /dev/scd7 hpe/CDROMs/Cdrom7 -o norock,uid=501,gid=10,suid,mode=0550
  ;;

start0)
 mount -t iso9660 /dev/scd0 hpe/CDROMs/Cdrom0 -o norock,uid=501,gid=10,suid,mode=0550
;;

start1)
 mount -t iso9660 /dev/scd1 hpe/CDROMs/Cdrom1 -o norock,uid=501,gid=10,suid,mode=0550
;;

start2)
 mount -t iso9660 /dev/scd2 hpe/CDROMs/Cdrom2 -o norock,uid=501,gid=10,suid,mode=0550
;;

start3)
 mount -t iso9660 /dev/scd3 hpe/CDROMs/Cdrom3 -o norock,uid=501,gid=10,suid,mode=0550
;;

start4)
 mount -t iso9660 /dev/scd4 hpe/CDROMs/Cdrom4 -o norock,uid=501,gid=10,suid,mode=0550
;;

start5)
 mount -t iso9660 /dev/scd5 hpe/CDROMs/Cdrom5 -o norock,uid=501,gid=10,suid,mode=0550
;;

start6)
 mount -t iso9660 /dev/scd6 hpe/CDROMs/Cdrom6 -o norock,uid=501,gid=10,suid,mode=0550
;;

start7)
 mount -t iso9660 /dev/scd7 hpe/CDROMs/Cdrom7 -o norock,uid=501,gid=10,suid,mode=0550
;;

stop)
       echo "Unmounting CD-ROMs.."

       umount /dev/scd0
       umount /dev/scd1
       umount /dev/scd2
       umount /dev/scd3
       umount /dev/scd4
       umount /dev/scd5
       umount /dev/scd6
       umount /dev/scd7
       ;;
```

```
  stop0)
        umount /dev/scd0
  ;;

  stop1)
        umount /dev/scd1
  ;;

  stop2)
        umount /dev/scd2
  ;;
  stop3)
        umount /dev/scd3
  ;;

  stop4)
        umount /dev/scd4
  ;;

  stop5)
        umount /dev/scd5
  ;;

  stop6)
        umount /dev/scd6
  ;;

  stop7)
        umount /dev/scd7
  ;;

  *)
        echo "Usage: rc.cdrom {start|stop|startn|stopn} where "n" is the CDROM drive ID"
        exit 1
esac

exit 0
--
```

- Make the rc.cdrom script executable by running "chmod r+x rc.cdrom" - Make the mount points for the CD- Changer's CDs:

```
                mkdir hpe/CDROMs/Cdrom0; mkdir hpe/CDROMs/Cdrom1; mkdir hpe/CDROMs/Cdrom2; mkdir hpe/CDROMs/Cdrom
                mkdir hpe/CDROMs/Cdrom4; mkdir hpe/CDROMs/Cdrom5; mkdir hpe/CDROMs/Cdrom6; mkdir hpe/CDROMs/Cdrom
```

- Change the permissions on the newly created dirs:

```
                chown 550 hpe/CDROMs/Cdrom*
                chgrp wheel hpe/CDROMs/Cdrom*
                chown hpe hpe/CDROMs/Cdrom*
```

- Edit the "/etc/rc.d/rc.local" file and add the following lines at the end:

```
                --
                #Run the cdrom mount script
                /etc/rc.d/rc.cdrom start
                --
```

33 Samba installation and configuration

Samba is the UNIX service for Microsoft Windows File and Print serving. The funny thing is, a well tuned Linux Samba server is a FASTER NT server than a well tuned NT server itself! As of Samba 2.0, it still doesn't offer full PDC/BDC support yet but it's coming in version 3.x.

* Please note that these installation docs are for Samba 1.9.x and might be somewhat different for a Samba 2.x distribution.

33.1 Determining what version you Samba you might have now

You should be running Samba 2.2.8a as all previous versions of Samba have serious security vunerabilities in dealing with issues like encrypted passwords, buffer overflows, etc. It is HIGHLY recommended that you make sure you are running 2.2.8a or better.

To find out what version you are running, do the following:

```
whereis smbd
/usr/sbin/smbd -V
```

33.2 Downloading and compiling Samba

Download the newest Samba source code /and/ the PGP signatures of the Samba archives from the URL given in 5 (Section 5). I recommend to put them into a directory such as `/usr/src/archive/samba`.

NOTE: These compiling installation instructions assume that you are running a Linux OS with a SHADOW password system. You really should be!

- First, verify that the PGP signature of the Samba source is ok (this step assumes you have GnuPG installed but not nessisarily be configured). `gpg -import samba-pubkey.asc`
  `cd /usr/src/archive/samba`
  `bzip2 -d samba-x.y.z.tar.bz2`
  `gpg -verify samba-x.y.z.tar.asc`

  Make sure it says "Good Signature" at the top. There might be some trust warnings but don't worry about that.

- Next, uncompress the .tar file: `tar -xvf samba-x.y.z.tar`

- Enter the new source directory
  `cd samba-x.y.z` `cd source`

- From here, Samba can be configured for various installation directories, different types of authentication, etc. To get an idea of what you can alter, run `./configure -help` if you want to mess with any of this. Basically, Samba offers a LOT of features now. It can be a WINS BDC (soon a full PDC), it supports client printer driver installation, database locking mechanisms, etc. - Please note that various Linux distributions (even different versions of the SAME distro) put the Samba binaries in different places. Samba does support the use of the emerging Linux file layout standard (FHS) but few comply today.

  - I recommend the use of the following tags until distros fully support FHS (good for Mandrake 7.2):

```
./configure --prefix=/usr --with-privatedir=/etc --with-lockdir=/var/lock/samba \
--with-configdir=/etc --with-smbmount --with-msdfs --with-smbwrapper
```

- Ok, compile it up:

```
make; make install
```

33.2.1 Specific Compiling issues:

For some of you, you might have received a compile error of

```
Compiling smbwrapper/wrapped.c with -fPIC
smbwrapper/wrapped.c:473: conflicting types for ‘utimes’
/usr/include/sys/time.h:112: previous declaration of ‘utimes’
```

This issue is due to the Samba code not properly recognizing that this code is conflicting with Linux's libraries. To fix this specific problem, disable the Samba version of the "utimes" code. To do this, edit the "src/smbwrapper/wrapped.c" file, goto line 472, and change the code from:

```
#ifdef HAVE_SYS_TIME_H
#include <sys/time.h>
#else
#include <time.h>
#endif

 int utimes(const char *name, const struct timeval *tvp)
{
    if (smbw_path(name)) {
        return smbw_utimes(name, tvp);
    }

    return real_utimes(name, tvp);
}
#endif
```

to the following:

```
#ifdef HAVE_SYS_TIME_H
#include <sys/time.h>
#else
#include <time.h>
#endif

/*
 int utimes(const char *name, const struct timeval *tvp)
{
    if (smbw_path(name)) {
        return smbw_utimes(name, tvp);
    }

    return real_utimes(name, tvp);
}
*/

#endif
```

Once this change is complete, run a "make clean" and re-run the "make"

For others Samba source code users:

- Older versions of Samba: - cd into the Samba directory and then "cd sources"

  - Edit the "Makefile"

  - Find the lines:

  "# The permissions to give the executables INSTALLPERMS = 0755"

  and change them to 0750"

  - Redhat users: find the following lines and un-#ed out the last two lines:

```
"# This is for PAM authentication. RedHat Linux uses PAM.
 # If you use PAM, then uncomment the following lines:
 # PAM_FLAGS = -DUSE_PAM
 # PAM_LIBS = -ldl -lpam"
```

  Ditto here:

```
"# FLAGSM = -DLINUX -DAXPROC -DFAST_SHARE_MODES
 # FLAGSM = -DLINUX -DFAST_SHARE_MODES
 # LIBSM ="
```

  Same here:

```
"# FLAGSM = -DLINUX -DNETGROUP -DALLOW_CHANGE_PASSWORD -DFAST_SHARE_MODES -D
 # LIBSM = -lnsl -lcrypt"
```

  - Save the changes and then run "make all; make install"

  - Security: Post from the Samba team on 11/20/98, you should do the following:

```
rm /usr/sbin/wsmbconf
chmod +t /var/spool/samba
```

33.3 Configuring the smb.conf file

The /etc/smb.conf file is the master file for Samba to both act as a server and as a client (connecting to remote SMB servers). So, edit the /etc/smb.conf file. If you need more information, run "man smb.conf" to read an exceptionally well written and detailed MAN page (it's much better than what you're probably thinking). For TrinityOS, this example shows how to create a few file shares and printer shares as well.

- Under the [Global] Section:

- Edit the "WORKGROUP" line to reflect the name of the workgroup you want

```
WORKGROUP = ACME123
```

- Edit the "server string" line to reflect the name of the machine

```
server string = TrinityOS Roadrunner Samba Server
```

- Edit the "hosts" allow line to ONLY reflect:

```
hosts allow = 192.168.0. 127.
```

- Make sure that printing is enabled:

```
printcap name = /etc/printcap
load printers = no
printing = bsd
```

- Make sure the GUEST account is disabled by having a ";" in the front of:

```
";  guest account = pcguest"
```

- For Windows 95/98/NT viewing, turn on "user level" security

```
"security = user"
```

- Windows XP, NT, Windows98, and patched Windows95 require ENCRYPTED SMB passwords. So, make sure you have the follow lines in your smb.conf file (or remove the ";"s if the lines are already there):

```
  encrypt passwords = yes
  smb passwd file = /etc/smbpasswd
```

- Since the Samba server and all clients are on the same LAN segment, add the following:

```
"socket options = IPTOS_LOWDELAY SO_RCVBUF=8192 SO_SNDBUF=8192"
```

- Since we have multiple Ethernet cards in the Roadrunner server, set the following:

```
"interfaces = 192.168.0.1/24 127.0.0.0/8"
```

- Add the line:

```
"bind interfaces only = true"
```

- Also set the following:

```
"remote announce = 192.168.0.255 "
```

- Allow Samba to be a subnet master browser

```
"local master = yes"
```

- Enable Samba to always win the Subnet Master Browser election

```
"preferred master = yes"
```

- Enable full Win95 login support:

```
"domain logons = yes"
```

- Fix Samba permissions so when you create a file/directory, the UNIX permissions are correct too!

```
"create mask = 0770"
"directory mask = 0750"
```

- **OPTIONAL / POSSIBLY an OLD config** Since my Samba server is only used by me, I can essentially disable file write locking on all shares. If you are going to have a lot of users editing the same file, you should NOT enable this option.

```
"fake oplocks = yes"
```

- **OPTIONAL** Since I have a CD-ROM changer on my machine, I don't need to enable file write locking on those file systems so I'll disable it here.

```
"veto oplock files = /home/hpe/CDROMs/Cdrom*"
```

- Set or verify the setting of follow shares for each user's home DIR and a central Hp Laserjet IIp printer.

* NOTE: The printer name CANNOT be any longer than -8 characters-!

```
[homes]
comment = Home Directories
# Making this NON-BROWSABLE gets rid of the duplicated "username" and
# "homes" shares
browseable = no
writable = yes
#  Allows only the current Samba user into their home directory
user = %S

[Hp_Lj2p]
printer = raw
```

```
comment = Hp LaserJet IIp on RoadRunner
path = /var/spool/samba
browseable = yes
# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes
print command = /usr/bin/lpr -b -r -PHp_Lj2p %s
lpq command = lpq -PHp_Lj2p
lprm command = lprm -PHp_Lj2p %j

[Epson_S]
printer = raw
comment = Epson Stylus 500 Color on RoadRunner
path = /var/spool/samba
browseable = yes
# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes
print command = /usr/bin/lpr -b -r -PEpson_S %s
lpq command = lpq -PEpson_S
lprm command = lprm -PEpson_S %j
```

- The /home/hpe directory is a common directory and SMB share for ALL users. Since ALL the files in this dir should be readable by all other users, I want all files/dirs to be created with the WHEEL group.

```
[hpe]
comment = Hpe
path = /home/hpe
read only = no
public = no
force group = wheel
--
```

33.4 Testing your smb.conf file

- Next, you need to test that your /etc/smb.conf file is correct. To do this, simply run the "testparm" program without any additional command line argments and it will check it for you and tell you everything it understands. Browse over this real quick but don't expect to understand much of it! Hehehe..

33.5 Loading Samba for the first time

- Now start up Samba, run

- Redhat:

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

- Slackware:

```
/usr/local/samba/bin/smbd -D
/usr/local/samba/bin/nmbd -D
```

33.6 Creating the smbpasswd file

- Lastly, we need to add your login to the Samba username file. Yes, it's separate from the normal /etc/password file. Though this is initially a pain, you can have it auto-syncronise with the UNIX password file (Not covered in the TrinityOS doc..yet) though it is covered in the Samba documentation.

— All of this is covered in /usr/doc/samba-*/ENCRYPTION.txt file —

- Ok, to create the /etc/smbpasswd file: run the following command:

```
cat /etc/passwd | mksmbpasswd.sh >/etc/smbpasswd
```

- Next, fix the permissions of the file:

```
chmod 500 /etc/smbpasswd
```

- With this command, all users defined in the /etc/passwd file will have a SMB entry put into the /etc/smbpasswd file. Please note that if desired, users can log in via a different SMB username/passwd than their Unix username/password. Please be aware that though the user is now defined in the smbpasswd file, the user will be LOCKED out until they actually CHANGE their SMB password. To do this, run the following command PER user:

```
                  smbpasswd johndoe
                  smbpasswd metarzan
      . . .
```

33.7 Specific Windows issues with Samba

- A few things to do on your Windows 95/NT box:

- One thing that you might not be used to doing is acutally logging into your Windows. You absolutely NEED to create a username AND a password on your Windows box to correspond to a username/password in the /etc/smbpasswd file on the Linux machine.

- Windows 95 - Use the Users Control Panel
- Windows NT - Use the User Manager

- You need to re-configure your Windows95 or WindowsNT servers to use the correct WORKGROUP (ACME123).

Windows 95 and NT: Set the Windows machine(s) to use a WORKGROUP of "acme123" (not a DOMAIN) and use "Share Level" protection.

NOTE: Verify that your Windows95/NT machine does NOT have the NetBEUI protocol installed. If it does, DELETE that protocol.

- Whew! Ok, the home stretch. Reboot your Windows boxes with the new WORKGROUP setting from the smb.conf file and when prompted, login with the configured **Windows** username and password from the above smbpasswd file. Once logged into the Windows machine, go to the "Network Neighborhood" and see if you see the ROADRUNNER server listed. If everything goes well, you should see your home UNIX directory!

So go for it and see if you can create, delete, move files, etc from File Explorer on your Windows machine. Cool huh?

33.8 Samba printing

If you want to do printing, check out 47 (Section 47)

** If you canot get Samba to run right, please read the Samba Diagnostic docs:

```
/usr/doc/samba-*/docs/DIAGNOSIS.txt
```

33.9 Having smbd load upon Linux reboot

- If everything went ok... Excellent! Congratulations! Now make sure that Samba or SMB is enabled to load upon boot.

- To do this, UN-DO all edits for SMB lines in 8 (Section 8) - Specifically, run the command:

- chkconfig –level=345 smb on

33.10 Listing and Mounting remote SMB shares locally on your Linux machine

On the flip side, you can mount your Windows95/NT shares onto your Linux box too. Cool huh!

- Assuming that everything worked above, you should be able get a list of shares from your Windows XP/2k/NT/Me/98/95 box, do:

```
"smbclient -L //your-windows-boxs-name -U johndoe"
```

When prompted for a password, enter in the same password that you use to log into your Windows95/NT machine. You should then see something like:

```
          Added interface ip=192.168.0.1 bcast=192.168.0.255 nmask=255.255.255.0
          Server time is Tue Jan 12 17:22:36 1999
          Timezone is UTC-8.0
          Password: <enter in the password of the Windows file share>
          Domain=[ACME123] OS=[Windows NT 4.0] Server=[NT LAN Manager 4.0]
          security=user

          Server=[your-nt-boxs-name] User=[] Workgroup=[ACME123] Domain=[]

          Sharename      Type      Comment
          ---------      ----      -------
          C$                 Disk
     IPC$           IPC      Remote IPC
```

- If the above step worked ok, you should be able to mount your Windows95/NT share directly onto your linux box. To do this, run the following:

mkdir /mnt/smb-c /usr/sbin/smbmount //your-nt-box-name/c$ /mnt/smb \ -o username=johndoe

34 PCMCIA services installation and configuration

- First.. make sure the PCMCIA cards you have are supported from a list available in the URL in 5 (Section 5). If your cards are supported (almost ALL are), download the newest version of software.

- Make sure your Linux kernel has TCP/IP support in it but you don't need to compile in any Ethernet card support. This is done by the PCMCIA modules. Tokenring is an exception to this rule.

- Uncompress the PCMCIA software in /usr/src or somewhere else you like

34.1 Compiling the PCMCIA tools

- run ./configure

- If you have the kernel sources install in /usr/src/kernel/linux, tell the ./configure script to use that to determine the kernel rev.

- I beleive that your card is a CardBus type so enable CardBus support.

- run make all

- run make install

+ Redhat: If this is for a Dell, this is how I would recommend you to configure your laptop. Note, you need to configure the network here and NOT from /etc/sysconfig. PCMCIA works in a totally different fashion than a standard NIC setup:

NOTE: You will need to include or exclude the rigth IRQs and IO ports for your machine.

34.2 Editing the PCMCIA configuration files

```
/etc/sysconfig/pcmcia   (for Redhat only)
--
PCMCIA=yes
PCIC=i82365
PCIC_OPTS="irq_list=3,5,9,10"
CORE_OPTS=
--
```

- All distributions: Edit the /etc/pcmcia/config.opts file:

```
--
#
# Local PCMCIA Configuration File
#
# System resources available for PCMCIA devices
#
include port 0x100-0x3ff, memory 0xc0000-0xfffff
#
# Extra port range for IBM Token Ring
#
include port 0xa20-0xa27
#
# Resources we should not use, even if they appear to be available
#
```

```
# Available IRQs for a Dell Latitude CP are 3,5,[9 is available if
#       MIDI support for the C4232 sound card is NOT enabled in
#       the kernel
#
# To be used for PCMCIA modem
include irq 3
# Used by interal DB9 serial port
exclude irq 4
include irq 5
# First built-in parallel port
exclude irq 7
include irq 9
# Used by PCMCIA Card controller
exclude irq 10
# Used by the CSS Sound Card
exclude irq 11
# PS/2 Mouse (trackpad)
exclude irq 12
# IDE Channnel #1
exclude irq 14
# IDE Channnel #2
exclude irq 15
#
# Options for loadable modules
#
# To fix sluggish network with IBM Ethernet adapter...
#module "pcnet_cs" opts "mem_speed=600"
#
# Options for Xircom Netwave driver...
#module "xircnw_cs" opts "domain=0x100 scramble_key=0x0"
--
```

/etc/pcmcia/networks.opts (for DHCP.. If you are using a static IP address.. turn OFF BOOTP here and enter in your IP address in the IPADDR field)

```
--
# Network adapter configuration
#
# The address format is "scheme,socket,instance,hwaddr".
#
# Note: the "network address" here is NOT the same as the IP address.
# See the Networking HOWTO.  In short, the network address is the IP
# address masked by the netmask.
#
case "$ADDRESS" in
*,*,*,*)
    # Transceiver selection, for cards that need it -- see 'man ifport'
    IF_PORT=""
    # Use BOOTP [y/n]
    BOOTP="y"
    # IP address
```

```
        IPADDR=""
        # Netmask
        NETMASK="255.255.255.0"
        # Network address
        NETWORK="1.2.0.0"
        # Broadcast address
        BROADCAST="1.2.255.255"
        # Gateway address
        GATEWAY="1.2.0.1"
        # Local domain name
        DOMAIN="ins.com"
        # Search list for host lookup
        SEARCH=""
        # Nameserver #1
        DNS_1=""
        # Nameserver #2
        DNS_2=""
        # Nameserver #3
        DNS_3=""
        # NFS mounts, should be listed in /etc/fstab
        MOUNTS=""
        # For IPX interfaces, the frame type (e.g., 802.2)
        IPX_FRAME=""
        # For IPX interfaces, the network number
        IPX_NETNUM=""
        # Extra stuff to do after setting up the interface
        start_fn () { return; }
        # Extra stuff to do before shutting down the interface
        stop_fn () { return; }
        ;;
    esac
    --
```

After you've done all this.. reboot your machine and while the BIOS is showing the memory, etc.. EJECT all your PCMCIA cards. After Linux has booted, login as root, and then hit ALT-F7 to check out all the logs.

- Insert one of your PCMCIA cards. Did it mount ok? (two high beeps?)
- To check, go back to your login TTY (Alt-F1) and run "ifconfig". Do you have an IP address?

- If everything is working ok, make sure that PCMCIA services is enabled upon boot.

- To do this, UN-DO all edits for PCMCIA lines in 8 (Section 8)

35 DHCPcd : Client DHCP for xDSL / Cablemodem users

All versions of DHCPcd prior to 1.3.22-p12 are vunerable to rogue DHCP servers. These hacked DHCP server could execute any commands on the vunerable DHCP client. Please make sure you are running 1.3.22-p12 or newer.

See 5 (Section 5) for some other excellent URLs on setting up DHCP clients

First, a quote from the TrinityOS firewall rule set about Linux DHCP clients:

```
--
# NOTE: Red Hat users of DHCP to get TCP/IP addresses (Cablemodems, DSL, etc)
#       will need to install and use a different DHCP client than the stock
#       client called "pump".  It should be noted that newer
#       versions of pump can run scripts upon lease bringup, renew, etc.  One
#       recommended DHCP client is called "dhcpcd" and can found
#       in Appendix A.
#
#       The stock Red Hat DHCP client doesn't allow the ability to have scripts
#       run when DHCP gets a TCP/IP address.  Specifically, DHCP delves out
#       TCP/IP addresses to its clients for a limited amount of time; this
#       called a "lease".  When a DHCP lease expires, the client will query the
#       DHCP server for a lease renewal.  Though the DHCP client will usually
#       get back its original TCP/IP address, this is NOT always guaranteed.
#       With this understood, if you receive a different TCP/IP address than
#       the IPCHAINS firewall was configured for, the firewall will block ALL
#       network access in and out of the Linux server because that was what it
#       was configured to do.
#
#       As mentioned above, the key to solve this problem is to use a DHCP
#       client program that can re-run the /etc/rc.d/init.d/firewall rule set
#       once a new TCP/IP address is set.  The new rule set will make the required
#       changes to the rule sets to allow network traffic from and to your new
#       TCP/IP address.
--
```

Another thing to note from the DHCPcd documentation:

```
--
In a case dhcpcd detects a change in assigned IP address it
will try to execute /etc/dhcpc/dhcpcd-interface.exe program.
The word <interface> is substituted by the actual interface name
like e.g. eth0. Caution: do not use /etc/dhcpcd-interface.exe
as a bootup script. It will not be executed if the assigned IP address
is the same as it was before reboot. The included sample
/etc/dhcpc/dhcpcd-eth0.exe will log the time of IP change
to /var/log/messages file.
--
```

- Note: 1. If you use TrinityOS's strong firewall rule set, you'll have to un-# out the "DHCP - Client" IPCHAINS or IPFWADM rule sets in both the Incoming and Outgoing rules to allow DHCP in through your EXTERNAL interface.

2. You will also have to execute the /etc/rc.d/rc.firewall when DHCP initial IP address or when it renews its IP address lease. Newer "dhcpcd" clients offer this functionality though not all of them do (such as "pump"). Be sure you use one that DOES have this function. It should be noted that newer versions of pump can run scripts upon lease bringup, renew, etc.

Here is a real quick intro on how to do this:

########

If you are running Mandrake 6.1, load up "vi" and go to /etc/sysconfig/network-scripts/ifup line 87. If you are running Redhat 6.x, edit the same file and do a search for "DHCP" (run the command "/DHCP" without the quotes).

You'll look for something like the following:

```
--
if [ -n "$DHCP" ]; then
    echo -n "Determining IP information for $DEVICE via dhcpcd..."
    if /sbin/dhcpcd -i $DEVICE -h $HOSTNAME ; then
        echo " done."
    else
        echo " failed."
        exit 1
--
```

You'll want to change it to something like the following (if it doesn't already look like this already).

```
--
if [ -n "$DHCP" ]; then
    echo -n "Determining IP information for $DEVICE via dhcpcd..."
    if /sbin/dhcpcd -H -D $DEVICE ; then
        echo " done."
    else
        echo " failed."
        exit 1
--
```

Next, you need to create a link to the firewall rule set for your given EXTERNAL interface:

ln -s /etc/rc.d/rc.firewall /etc/dhcpc/dhcpcd-*EXTIF*.exe

Replace the "*EXTIP*" for the name of your external interface. For example, if your external interface is "eth0", it would be:

ln -s /etc/rc.d/init.d/firewall /etc/dhcpc/dhcpcd-eth0.exe

That's it! Now when the /sbin/ifup script or dhcpcd programs are called, they will get their IP address and then run the firewall rule set automatically.

- Other DHCP tricks: One thing that DHCP does -not- give out is DNS search lists. To me, thats a pretty big bummer. But you can fake it with a script executed from the dhcpcd-*.exe file once you get your initial DHCP lease. Please note that you'll have to create a master dhcpcd-eth*.exe file that runs both the rc.firewall script AND the DNS-search trick if you want all this functionality in one place.

  /etc/dhcpc/dhcpcd-eth0.exe

```
#!/bin/bash

search=`grep -c -e acme /etc/resolv.conf`
#echo $search
if [ "$search" != "0" ]; then
  mv -f /etc/resolv.conf /etc/resolv.conf.old
```

```
  echo "domain acme123.com" > /etc/resolv.conf
  grep -v -e "search" -e "domain" /etc/resolv.conf.old >> /etc/resolv.conf
  echo "search acme123.com backupacme.com" >> /etc/resolv.conf
fi
```

Please note: Once you setup this DNS-search hack, things might not work. To get things running, delete the /etc/dhcpc/dhcpcd-eth0.info and .cache (this example is for eth0) files. Then run "ifdown eth0" and then "ifup eth0".

36 UPS: Complete UPS Backup & Graphing support for APC UPSes

36.1 The state of the software

Today, APC UPSes are fully supported by both OpenSource and APC proprietary software for Linux. Overall, both versions do their job well but they don't completely overlap in features and flexibility. The APC version is short, sweet, and does 90% of everything you could ever want. On the flip side, the OpenSource versions allow for remote shutdown of internal LAN-based PCs, etc. Here is a breakdown of the PROs/CONs of both pacakges:

OpenSource APCUPSd:

- + Shutdown of machines (Linux, Windows, etc.) via basic TCP/IP connectivity
- + Powerful flexibility upon UPS power events, etc.
- + Allows for ultra-fine logging
- + Pretty simple to setup
- - Not compatible with a controlling APC Powerchute daemon
- - Does not support UPS battery "Run-Time Calibration" (fixed soon)
- - GUI tools are present but not very feature-rich

APC Powerchute Plus (NOT the Business Edition - free but proprietary):

- + Simple to setup
- + Has a nice GUI to configure the UPS
- + Allows to conduct UPS "runtime calibration"
- + Fairly powerful mechanism upon UPS power events, etc.
- - Cannot directly use TCP/IP networks to signal other machine shutdowns.. even if they are ALL running APC's Powerchute software. You have to buy APC's SNMP hardware card to support this feature. Update: It seems the Business Edition will allow for this and it's free for 5-nodes.
- - Logging isn't very granular
- - requires the Xwindows GUI for configuration / text interface configuration was removed
- - Networking uses 255.255.255.255 broadcast packets on all interfaces and the binding of what interfaces to use is NOT configurable. The explict use of internal hostnames under the "HostName" doesn't help

- - GUI will let you select Fahrenheit vs. Celcius and the display of "Battery Capacity" vs. "Battery Voltage". Unfortunately, the .dat log files will only show Celcius and Batt. voltage.

This TrinityOS chapter covers:

- Installation and setup of the OpenSource APCUPSd software
- Full scripts for paging, emailing, and logging
- A cool script that graphs each day's power conditions in an emailed .PDF

One difference that should be mentioned again is that the official APC Powerchute software for Linux is NOT compatible with MS Windows UPS clients written by APC. This means that you cannot use your internal LAN to shutdown other MS Windows machines in addition to your Linux machine.

Currently, these docs only cover the installation of the OpenSouce "apcupsd" tool from both RPM and tar.gz form. If there is enough interest, I can also describe the setup of APC Powerchute software too. I still recommend the OpenSource version (it DOES shutdown other machines running OSes like Windows, etc.). Think modular. :-)

36.2 Installing and Using APC's Powerchute

If you still want to run Powerchute software over the APCUPSd program, I recommend that you:

- NOT enable "networking" support Powerchute doesn't have the ability to configure which interfaces the software binds to. Because of this, you'll be spamming Powerchute broadcast packets (yes, 255.255.255.255 packets) to /all/ interfaces on server. This is lame and is APC's issue.

  If you are running a strong firewall (you should be), the FW will block the xpowerchute GUI from finding your local powerchute daemon.

  What to do? If you don't need to monitor other remote Powerchute daemons from this server, just don't enable networking when installing Powerchute. If you've already installed powerchute, simply edit the /usr/lib/powerchute//powerechute.ini file and change the line:

```
        UseTCP = Yes
```

  to

```
        UseTCP = No
```

  Now simply restart the daemon:

```
        /etc/rc.d/init.d/upsd restart
```

  and now try running /usr/lib/powerchute/xpowerchute. Hopefully it will run without issue.

36.3 Installing APCUPSd

Ok..

- Download the newest APCUPSd found in 5 (Section 5)

- If you downloaded the RPM, install it with the command:
  - rpm -Uvh apcupsd-x.y-z.i386.rpm
- If you downloaded the tar.gz file, uncompress it, configure it, and compile it:
  - tar xzvf apcupsd-3.x.y-z.tar.gz
  - cd apcupsd-x.y-z
  - ./configure –enable-powerflute
  - make
  - make install

- Next, fix its permissions:

```
                chmod 750 /sbin/apcupsd
```

36.4 Configuring APCUPSd for logging and paging

Redhat:

- Make sure that /etc/rc.d/rc3.d/S20apcupsd exists

Next, edit /etc/apcupsd/apcupsd.conf and make the following changes. Please note that you need to alter the example to better match your environment.

/etc/apcupsd/apcupsd.conf

```
UPSCABLE smart
UPSTYPE smartups
DEVICE /dev/ttyS0
LOCKFILE /var/lock

BATTERYLEVEL 10
MINUTES 0
TIMEOUT 0
ANNOY 300
PROCFS 5
ANNOYDELAY 60
NOLOGIN disable
KILLDELAY 0

#Set only to on if you plan to shutdown other machines via a TCP/IP network
NETSERVER off

EVENTSFILES /var/log/apcupsd.events
```

```
STATTIME 0
STATFILE /var/log/apcupsd.status
LOGSTATS off

#Log UPS stats once a second
DATATIME 1

#Newer APCUPSd programs no longer log directly to a data file.
#  The newer versions now log ONLY to SYSLOG
FACILITY local0

SENSITIVITY H
WAKEUP 180
BEEPSTATE L
SELFTEST 336

UPSCLASS standalone
UPSMODE disable
NETACCESS false
--
```

The next step is to configure SYSLOG to support the new APCUPSd logging system (APCUPSd no longer logs directly to a specified file). Edit the /etc/syslog.conf file and add the following line:

/etc/syslog.conf

```
local0.*                        /var/log/apcupsd.data
```

Ok, so this is nice and all but the common SYSLOG setup in Linux will also send ALL log messages to other files as well. There is no need to mess up these other files with the intentionally chatty UPS log stats so I recommend to modify other "*.*" lines to exclude these once-a-second UPS stats info. Please edit all the syslog lines that apply but this example should cover it:

/etc/syslog.conf

```
*.*;local0.!info                               /var/log/syslog

*.info;mail.none;authpriv.none;local0.!info    /var/log/messages
```

Once this is all setup, you should activate both the new log file and the new SYSLOG system:

Redhat:

- touch /var/log/apcupsd.data
- chmod 600 /var/log/apcupsd.data
- /etc/rc.d/init.d/syslog restart

Slackware:

- touch /var/log/apcupsd.data

- chmod 600 /var/log/apcupsd.data
- kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk ’{print $2}‘‘

Optional stuff: Paging users when power events occur:

- In addition to the system gracefully shutting things down, some people might want the system to notify them via a pager, cellphone, etc. NOTE: If you don’t want to enable the paging feature, simply skip this section.

  NOTE 2: Change the pager email address to reflect both your pager ID and pager server

  NOTE 3: Please notice that the old APCUPSd /usr/local/sbin/apcupsd-* scripts have now been replaced the the master "/etc/apcupsd/apccontrol" script. Now you only need to edit this file to do what you want.
- Here are only some IDEAS on what to log but please edit the file and make the appropreiate substitutions to your tastes:

/etc/apcupsd/apccontrol

```
emergency)
   wall "Emergency Shutdown. Possible battery failure on UPS ${2}."
   echo "Emergency! Batteries have failed on UPS ${2}. Change them \
NOW" | /bin/mail 1234567@skytel.com
   ${SHUTDOWN} -h now "apcupsd emergency shutdown"
;;

onbattery)
   wall "Power failure on UPS ${2}. Running on batteries."
   /usr/bin/logger "Power failure on UPS ${2}. Running on batteries."
   echo "Power failure on UPS ${2}. Running on batteries." \
| /bin/mail 1234567@skytel.com
;;
```

Now, fix the permissions on the files:

- chmod 700 /usr/local/sbin/apcupsd-*

Finally, you need to TEST the new UPS setup:

- Connect up the UPS control cable to the UPS, plug-in the UPS to the wall outlet but DO NOT HAVE THE COMPUTER CONNECTED TO THE UPS QUITE YET.
  - First, change the /etc/apcupsd.conf variable:

```
TIMEOUT 120
```

    The reason to do this is to be able to test the setup quickly without draining the battery.
  - Start the apcupsd process by typing in:

```
/sbin/apcupsd -f /etc/apcupsd/apcupsd.conf
```

36.5 Testing your new UPS setup

- To make sure things are perfect, just pull the plug on the UPS. ;-) Pull the power from the UPS and wait 2 minutes. Make sure that the system shuts down ok and then powers OFF. Please note that APC SmartUPS (not BackUPS or BackUPS Pro models) then remove the power from the computer(s) until the main wall AC power is back and the UPS is somewhat recharged. Other UPS will just simply come back on when the power is returned. The main problem with this is if the power goes back out, the UPS might not have enough power to gracefully shut the machine back down.
- If the UPS doesn't react as you expect, fix it NOW. Trust me on this one. A misconfigured UPS can be an absolute NIGHTMARE and ultimately cost your PCs or even your dewelling (I had a UPS literally blow up on me - see below).
- Now, re-plug in the UPS back into wall AC power and make sure that the system powers up ok and the file systems mount cleanly.
- If everything is ok, change the "TIMEOUT" parameter back to "0". Shut the computer down and plug it's power cord into the UPS's output.
- Make sure the PC re-powers back up (if this machine is a Internet server or not power-up if you don't care). If it doesn't do what you want to do, look in the "Advanced" sections of your PC's BIOS for something like "System start upon AC powerloss: YES".

36.6 Graphing the UPS stats results each day

As mentioned above, I once had a UPS that lost control of the charging circuit and and nearly burned down my house. Ever since then, I felt that I needed to always monitor the envirtonmentals of my UPS. Hopefully this will help prevent this catastrophe from ever happening to me again.

The following script will take the previous day's APCUPSd or APC Powerchute logs and create a high quality multicolor graph in PDF format. Not only that but the PDF is emailed to you via CRON every night. Check out `<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS-security/var/log/ups-log-jun24.pdf>` to see an example PDF of my terrible day. Specifically look at the temperature line and imagine the worst sulfur smell you could imagine! Overall, I got lucky!

Please also notice that this script has a BUNCH of pre-installed software requirements but most machines should have this already installed. Please see the comments in the script below for full details. Like any shell script, you can change things around to better fit your needs.

Download the script directly: Within the `<http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz>` archive

or

Just the file:

Powerchute: `<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS-security/usr/lib/powerchute/powerchute-generate-ups-graph.sh>`

APCUPSd: `<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS-security/usr/local/sbin/apcupsd-generate-ups-graph.sh>`

- Currently, this script uses relative paths which is bad (sorry.) Once I get a chance, I'll fix this. Until then, this file should be placed in /usr/local/sbin/ (APCUPSd users) or /usr/lib/powerchute (Powerchute users).

Here is the script for Powerchute:

<TrinityOS powerchute-generate-ups-graph.sh START>

```
#!/bin/sh

# TrinityOS - powerchute-generate-ups-graph.sh
# written by David Ranch
# v1.50
#
#   Changes
#   -------
#     1.5 - Fixed a long standing OCTAL conversion error
#     1.2 - Added some additional debugging options
#     1.1 - Updated to reflect support for both APCUPSd and Powerchute
#           and noted possibly Mutt attachment issues
#     1.0 - Original version
#
#
# This script takes the output from APC's Powerchute for Linux and
# both graphs it and emails it to the administrator.
#
# If you are running the OpenSource APCUPSd tool, please use the
#    apcupsd-generate-ups-graph.sh script available in TrinityOS.
#
# NOTE: This script requires:
#        - Powerchute for Linux installed and running properly
#        - bash
#        - awk
#        - gnuplot
#        - ps2pdf (ghostscript)
#        - mutt
#
# NOTE#2:  APC Powerchute v4.5.2 has a log file size limitation of
#          750k per the powerchute.ini file but APCUPSd doesn't have
#          this limitation.  Because of this Powerchute limit,
#          I've found that you CANNOT sample anything faster than
#          say 7 seconds.  Obviously, this isn't very granular.
#          If 7 seconds is just enough, you MUST run this script
#          around midnight or the script will fail due to missing
#          data.


#Local vars
#
#Machine running the UPS software
HOST="roadrunner"
#Who the resulting email should goto
ADMIN="johndoe@acme123.com"

# ================================================================
```

```
clear
cd /usr/lib/powerchute

#date setup
MONTH=‘date +%m‘
DAY=‘date +%d‘
YES=$((10#$DAY-1))
YEAR=‘date +%y‘
YESTERDAY="$MONTH/$YES/$YEAR"

#DEBUG - enable and change the DAY line to graph a specific day
#        and make sure you
#DAY=20
#YES=$(($DAY-1))
#echo -e "\n\nDEBUG: Graphing $YESTERDAY\n\n"


#Need to remove the commas and such
#  This is setup to manipulate Powerchutes logs.  You must make slight
#  changes to this to handle APCUPSds logs (it has a few more fields)
#  Feel free to email me if you need a hand.
#

echo -e "Beginning process to create graph for: $YESTERDAYi\n"
echo "Filtering original powerchute.dat file.."
cat powerchute.dat | \
  awk -F , ’{print $1" "$2" "$3" "$4" "$5" "$6" "$7" "$8" "$9}’ \
  > filtered-powerchute.dat

#Ok, now create the gnuplot command file
echo "set title \"$HOST $YESTERDAY APC Powerchute Log\"" > generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xlabel \"Date\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set ylabel \"Absolute number\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set timefmt \"%m/%d/%y %H:%M:%S"\" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xdata time" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xrange [ \"$MONTH/$YES/$YEAR\":\"$MONTH/$DAY/$YEAR\" ]" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set terminal postscript" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set terminal postscript color" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set terminal postscript solid" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set output \"/tmp/ups-log-$MONTH$YES$YEAR.ps\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot

#This is for Powerchutes logs.  If you are using APCUPSd, you will need
#to make slight changes here as the order is a little different and APCUPSd
#also has a few extra files too.
echo "plot \"filtered-powerchute.dat\" using 1:3 title ’LineMIN’ with lines, \\" >> generate-apc-graph-$MONTH$YES
echo "  \"filtered-powerchute.dat\" using 1:4 title ’LineMAX’ with lines, \\" >> generate-apc-graph-$MONTH$YES$YE
echo "  \"filtered-powerchute.dat\" using 1:5 title ’OutV’ with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR.
echo "  \"filtered-powerchute.dat\" using 1:6 title ’BattV’ with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR
echo "  \"filtered-powerchute.dat\" using 1:7 title ’LineFREQ’ with lines, \\" >> generate-apc-graph-$MONTH$YES$Y
```

```
echo "  \"filtered-powerchute.dat\" using 1:8 title 'UPSload' with lines, \\" >> generate-apc-graph-$MONTH$YES$YE
echo "  \"filtered-powerchute.dat\" using 1:9 title 'UPStemp' with lines" >> generate-apc-graph-$MONTH$YES$YEAR.g

echo "Deleteing old ps and pdf files.."
#rm -f /tmp/ups-log*.ps /tmp/ups-log*.pdf

echo "Creating files.."
gnuplot generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo " - done creating files"

echo "Creating /tmp/ups-log-$MONTH$YES$YEAR.ps.."
ps2pdf /tmp/ups-log-$MONTH$YES$YEAR.ps
rm -f /tmp/ups-log-$MONTH$YES$YEAR.ps
mv -f ups-log-$MONTH$YES$YEAR.pdf /tmp

echo "Cleaning up.."
#rm -f filtered-powerchute.dat
rm -f generate-apc-graph-$MONTH$YES$YEAR.gnuplot

# NOTE: If the emailed PDF seems to be corrupt, make sure that you
#       have the /etc/mailcap file installed
#
echo "Emailing graph.."
echo "Results for $MONTH$YES$YEAR" | \
  mutt -a /tmp/ups-log-$MONTH$YES$YEAR.pdf \
  -s "$HOST UPS graph for $MONTH$YES$YEAR" $ADMIN

#Uncomment this out once you are SURE things are working.  If things
#are NOT working, make sure this file exists if not check that you
#have all the required tools installed, etc.
#
#rm -f /tmp/ups-log-$MONTH$YES$YEAR.pdf
```

<TrinityOS powerchute-generate-ups-graph.sh STOP>

Here is the script for APCUPSd:

<TrinityOS apcupsd-generate-ups-graph.sh START>

```
#!/bin/sh

# TrinityOS - apcupsd-generate-ups-graph.sh
# written by David Ranch
# v1.50
#
#   Changes
#   -------
#     1.5 - Fixed a long standing OCTAL conversion error
#     1.2 - Added some additional debugging options
#     1.1 - Updated to reflect support for both APCUPSd and Powerchute
#           and noted possibly Mutt attachment issues
#     1.0 - Original version
```

```
#
# This script takes the output from APCUPSd for Linux and
# both graphs it and emails it to the administrator.
#
# If you are running APC"s Powerchute for Linux,  please use the
#    powerchute-generate-ups-graph.sh script available in TrinityOS.
#
# NOTE: This script requires:
#        - APCUPSd for Linux running properly (doc'ed in TrinityOS)
#        - bash
#        - awk
#        - gnuplot
#        - ps2pdf (ghostscript)
#        - mutt
#


#Local vars
#
#Machine running the UPS software
HOST="Roadrunner"
#Who the resulting email should goto
ADMIN="johndoe@acme123.com"

# ==================================================================

clear

#Enable this line if you run APCUPSd
cd /var/log

#date setup
MONTH=`date +%b`
DAY=`date +%d`
YES=$((10#$DAY-1))
TOM=$((10#$DAY+1))
YEAR=`date +%y`
YESTERDAY="$MONTH/$YES/$YEAR"

#DEBUG - enable and change the DAY line to graph a specific day
#        and make sure you
#DAY=20
#YES=$(($DAY-1))
#echo -e "\n\nDEBUG: Graphing $YESTERDAY\n\n"


# Need to remove the commas and such
#
#  This script manipulates APCUPSd logs.  If you are running Powerchute,
#  please use the Powerchute script shown above instead
```

```
#

echo -e "Beginning process to create graph for: $YESTERDAY\n"
echo "Filtering original apcupsd.data file.."
cat apcupsd.data | grep -v "succeeded" | grep -v "repeated" | \
  awk '{print $1" "$2" "$3" "$6}' | \
  awk -F , '{print $1" "$2" "$3" "$4" "$5" "$6" "$7" "$8" "$9" "$10}' \
  > filtered-apcupsd.data


#Ok, now create the gnuplot command file
echo "set title \"$HOST $YESTERDAY APC APCUPSd Log\"" > generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xlabel \"Date\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set ylabel \"Absolute number\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set timefmt \"%b %d %H:%M:%S"\" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xdata time" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot

#debug
#echo "set xrange [ \"$MONTH $DAY\":\"$MONTH $TOM\" ]" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set xrange [ \"$MONTH $YES\":\"$MONTH $DAY\" ]" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot

#Disable the following FOUR lines to display the graph in a Xwindow
echo "set terminal postscript" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set terminal postscript color" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set terminal postscript solid" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo "set output \"/tmp/ups-log-$MONTH$YES$YEAR.ps\"" >> generate-apc-graph-$MONTH$YES$YEAR.gnuplot

#This is for APCUPSd logs.
echo "plot \"filtered-apcupsd.data\" using 1:4 title 'LineMIN' with lines, \\" >> generate-apc-graph-$MONTH$YES$Y
echo "  \"filtered-apcupsd.data\" using 1:5 title 'LineMAX' with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR
echo "  \"filtered-apcupsd.data\" using 1:6 title 'OutV' with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR.gn
echo "  \"filtered-apcupsd.data\" using 1:7 title 'BattV' with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR.g
echo "  \"filtered-apcupsd.data\" using 1:8 title 'LineFREQ' with lines, \\" >> generate-apc-graph-$MONTH$YES$YEA
echo "  \"filtered-apcupsd.data\" using 1:9 title 'UPSload' with lines, \\" >> generate-apc-graph-$MONTH$YES$YEAR
echo "  \"filtered-apcupsd.data\" using 1:10 title 'UPStemp' with lines" >> generate-apc-graph-$MONTH$YES$YEAR.gn

echo "Deleteing old ps and pdf files.."
rm -f /tmp/ups-log*.ps /tmp/ups-log*.pdf

echo "Creating files.."
gnuplot generate-apc-graph-$MONTH$YES$YEAR.gnuplot
echo " - done creating files"

echo "Creating /tmp/ups-log-$MONTH$YES$YEAR.ps.."
ps2pdf /tmp/ups-log-$MONTH$YES$YEAR.ps
rm -f /tmp/ups-log-$MONTH$YES$YEAR.ps
mv -f ups-log-$MONTH$YES$YEAR.pdf /tmp

echo "Cleaning up.."
rm -f filtered-apcupsd.data
```

```
rm -f generate-apc-graph-$MONTH$YES$YEAR.gnuplot


# NOTE: If the emailed PDF seems to be corrupt, make sure that you
#       have the /etc/mailcap file installed
#
echo "Emailing graph.."
echo "Results for $MONTH$YES$YEAR" | \
 mutt -a /tmp/ups-log-$MONTH$YES$YEAR.pdf \
 -s "$HOST UPS graph for $MONTH$YES$YEAR" $ADMIN

#Uncomment this out once you are SURE things are working.  If things
#are NOT working, make sure this file exists if not check that you
#have all the required tools installed, etc.
#
#rm -f /tmp/ups-log-$MONTH$YES$YEAR.pdf
```

<TrinityOS apcupsd-generate-ups-graph.sh STOP>

Next, make the script executable:

```
chmod 700 /usr/lib/powerchute/powerchute-generate-ups-graph.sh
```

OR

```
chmod 700 /usr/local/sbin/apcupsd-generate-ups-graph.sh
```

Ok.. to get things running once a night, we need to use CRON:

- Redhat: To have the script run once a night, create a symbolic link in the SysV-style cron setup:

  ```
  ln -s /usr/lib/powerchute/powerchute-generate-ups-graph.sh \
    /etc/cron.daily/powerchute-generate-ups.graph.sh
  ```

  OR:

  ```
  ln -s /usr/lib/powerchute/apcupsd-generate-ups-graph.sh \
    /etc/cron.daily/apcupsd-generate-ups.graph.sh
  ```

  If you are using APC's Powerchute for Linux, you really need to have the powerchute-generate-ups-graph.sh script run EXACTLY at midnight. The reason for this is that Powerchute's logs have a maximum file size (see comments the script above) and the way TrinityOS configures Powerchute.. you will MAX this file limit out every day.

  To ensure things run on time, change the line in /etc/crontab to start the cron.daily script at 12:04 instead of 4:04:

  ```
          02 4 * * * root run-parts /etc/cron.daily
  ```

  to

```
        02 0 * * * root run-parts /etc/cron.daily
```

Once that is fixed, restart CRON by running:

```
        /etc/rc.d/init.d/crond restart
```

Ok.. one last thing: With such an agreesive logging schedule, APCUPSd can create VERY large files (805k per day). Powerchute doesn't have this issue since it automatically rotates the logs once the file hits 750k. This limit is both nice but also VERY limiting. With APCUPSd, I recommend to rotate the logs at LEASE every week. To do this, APPEND the following lines to the end of the /etc/logrotate.d/syslog file (Redhat only):

/etc/logrotate.d/syslog

```
/var/log/apcupsd.data {
        rotate 5
        weekly
        postrotate
        /usr/bin/killall -HUP syslogd
        endscript
}
```

That's it. Enjoy!

37 Apache WWW Server

Sorry this is so brief but setting up a simple Apache WWW server is very easy. But, configuring all of the advanced features is WAY out of the scope of this doc.

- Download the newest version of the standard Apache or SSL-encrypted WWW server for Linux from the URL in 5 (Section 5)

- Install the new apache software:

Redhat: rpm -Uvh apache-1.2.6-5.i386.rpm

Slackware: tar -xzvf apache_1.2.6.tar.gz

- Now, edit your WWW pages in the following directories based upon your Linux distribution

Redhat: /home/httpd/html

- Upon the fact that the WWW server runs fine, re-enable HTTPD upon boot.

- To do this, UN-DO all edits for HTTPD lines in 8 (Section 8)

- Also don't forget to re-enable HTTPD log rotation if you disabled it towards the end of 9 (Section 9).

- If you want to be able to directly FTP files to the /home/httpd/html directory, you need to make sure the given logins and the Apache html dir has proper group permissions:

- edit /etc/passwd and in the 4th field delimenated by ":"s, change the GID or GroupID to "4" for ALL people that should be able write to the global HTML dir.

```
                                i.e.  dranch:x:500:4::/home/dranch:/bin/bash
```

- Next, fix the permissions of the /home/httpd/html dir

```
                                chgrp -R adm /home/httpd/html
                                chmod 775  /home/httpd/html
                                chmod 764 /home/httpd/html/*
```

38 Tripwire file monitoring [Not finished yet]

Tripwire is a file monitoring application that can be configured to notify the administrator if any files have been altered. With a system like this in place, administrators will have a clear picture of what files have been changed during:

- file system corruption
- accidental changes
- hacker intrusion

- First, download the tripwire software from 5 (Section 5) and put it into a temporary directory

- Next, decompress it:

```
                tar -xzvf tripwire-*.tar.Z

                tar -xvf T1.2.tar
```

- Now go into the new tripwire-1.2 source dir

- Edit Makefile

out CC = cc

and un#ed out

CC = gcc

out LEX = lex

Un#ed out

LEX = flex

out YACC = yacc

un#ed out:

YACC = bison -y

39 Backing up the new system Linux to a CD-R

- Download mkisofs from the URL in 5 (Section 5)

- Uncompress the archive

```
tar -xzvf mkisofs-1.11.3.tar.gz
```

- Now do the following:

```
./configure
make
make install
```

- Next, assuming that you have enough drive space on your local HD (run a "df" to check) and you have at LEAST 16MB of RAM (per the mkisofs docs. Trust me, its true), do the following:

```
cd /
mkisofs -o /tmp/TrinityOS-101098.iso -a -L -R -V TrinityOS .
```

This will create a ISO image in /tmp which will include all files (-a), allow files to start with a "." (-L), enable RockRidge extentions to support EXT2 file permissions (-R), give the ISO image a volume name of "TrinityOS" and backup the files from the current directory (/).

40 NFS (Network File System) File sharing

NFS is one of the original network-based file sharing systems that was developed by Sun Corporation. NFS is one of the many services that Sun developed for their network architechure called RPC or Remote Procefure Call. The various other RPC services offer some amazing functionality such as remote quotas, remote WALLing people, etc. but for now, we will concentrate on NFS.

NFS is considered in many circles to be UN-SECURE. Because of this, few system admins are willing to run it in fear of losing security. Though there are many truthful aspects to this statement, NFS can be made to be more secure and limit its exploitability. To reduce any NFS-related security issues, take the following to heart:

40.1 NFS Security:

1. Setup a strong packet firewall as shown in TrinityOS or setup a statefully-inspected firewall to protect your NFS server from unauthorized machines (expensive but the ultimate). See below on how to change the TrinityOS IPCHAINS or IPFWADM rule sets to allow in external NFS traffic

2. Setup TCP wrappers as shown below

3. Only allow NFS access from specific NFS clients via the firewall, TCP wrappers, and the /etc/exports file.

4. Even if a NFS hacker got in, they CANNOT traverse to other non-NFS'ed file systems . So, put all your NFS-sharable data on one specific file system. With this in place, you greatly limit your NFS risk.

40.2 Note about Linux NFS performance:

Linux's NFS support somewhat slow. The reason for this is because the NFS support in Linux's 2.0.x and 2.1.x kernels are in what is called "user space". Because of this, the kernel doesn't have direct control and thus all NFS data transfers have to go through an excessive number of operating system layers. Fortunately, the upcoming Linux 2.2.x kernels will support NFS in "kernel space" which should bring its performance on par with many other UNIXes including the likes of Free/Open/Net-BSD.

There are several NFS optimizations that you can make to NFS but many of them can make NFS unstable. Once I have more time, I will document these tweaks but until then, the LDP's NFS-HOWTO located in /usr/doc/HOWTO or your local LDP mirror documents all this very well.

Down to it...

—

- First, you need to make sure that you compiled in NFS support into the Linux kernel as shown in 12 (Section 12). If you didn't, you will need to re-follow that section, enable NFS, compile the kernel, and reboot with the new kernel.

- Second, you need to specify what files on the NFS server you want to make available to remote NFS clients. To do this, create/edit the following file. All additional NFS shares should be put on their own line:

```
/etc/exports
--
#NFS exports file
#
#In a pinch to backup a whole remote file system
/              192.168.0.2(rw,no_root_squash)
/home/hpe      192.168.0.2(rw) 192.168.0.4(ro) 192.168.0.10(ro,nosuid,noexec)
--
```

In this configuration file, the first line will allow host 192.168.0.2 full read/write permissions to ALL files (root see's all) on the entire system. The second line will allow the 192.168.0.2 to both READ/WRITE to all files on the NFS server located in "/home/hpe" but only allow 192.168.0.4 READ ONLY access. 192.168.0.10, on the other hand, can only READ this volume and cannot RUN any programs from this NFS share.

In addition to all this, this config only allows users at the various IPs access files and directories which they ALREADY have UNIX permission to. NFS still enforces permissions based on the UserID and GroupID of the user.

There are a LOT of other options here that you might want to run (allow in a whole wildcarded domain, etc.) so check out the well written man page (man exports) or NFS-HOWTO.

- Next, Linux's NFS supports TCP Wrappers. Because of this, you need to configure TCPD to allow all of your desired clients to connect via NFS.

```
/etc/hosts.allow
--
ALL: 192.168.0.2

portmap: 192.168.0.4/255.255.255.255
--
```

What this means is that host 192.168.0.2 is allowed to access ALL services on the server where as host 192.168.0.4 is ONLY allowed to connect via the RPC Portmapper service.

- Another area of security involves the IPFWADM and/or IPCHAINS packet firewalls. My default IPCHAINS and IPFWADM policies allow *ANY* type of traffic to hit the Linux server from the internal NIC but *REJECT* most types of traffic from the Internet. I would highly recommend that you do this as well. If you have specific needs to enable NFS on your Internet link, you will need to edit your IPCHAINS/IPFWADM rule file and allow:

```
Port 111  [TCP and UDP] - for the RPC portmapper
Port 635  [UDP]         - for the NFS mounter
Port 2049 [TCP and UDP] - for NFS
```

For example, change the IPFWADM rule sets for your various EXPLICTITLY allowed-in hosts from 10 (Section 10) to add the above TCP and UDP ports:

Incoming traffic:

```
#secure1.host.com
/sbin/ipfwadm -I -a accept -W $extif -P tcp -S $securehost/32 -D $extip ftp ftp-data ssh pop-3 63
# NFS support
/sbin/ipfwadm -I -a accept -W $extif -P udp -S $securehost/32 -D $extip 111 635
```

Outgoing traffic:

```
#secure1.host.com
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 -D $securehost/32 ftp ftp-data ssh $unpr
#NFS traffic
/sbin/ipfwadm -O -a accept -W $extif -P tcp -S $extip/32 635 -D $securehost/32
/sbin/ipfwadm -O -a accept -W $extif -P udp -S $extip/32 111 2049 -D $securehost/32
```

- Next, you need to load the RPC Portmapper, mountd, and NFS daemons. You can load them by hand by running the following commands:

Manually:

```
--
/usr/sbin/portmap
/usr/sbin/rpc.mountd
/usr/sbin/rpc.nfsd
--
```

Redhat:

```
--
/etc/rc.d/init.d/portmap start
/etc/rc.d/init.d/nfs start
--
```

If you want to run these services permanently, go back to the "Initial System Security Section" 8 (Section 8) and undo all NFS, RPC, and Portmapper-related changes for your specific Linux distribution.

- Ok, NFS should be running now. Just to make sure, run the following command and verify it's output:

```
[root@roadrunner iana]# rpcinfo -p

   program vers proto   port
    100000    2   tcp    111  rpcbind
    100000    2   udp    111  rpcbind
    100005    1   udp    635  mountd
    100005    2   udp    635  mountd
    100005    1   tcp    635  mountd
    100005    2   tcp    635  mountd                100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

- Next, from the client machine that you want to mount a given NFS share, run

```
showmount 192.168.0.1
```

And see if you get a list of NFS shares.

- For the home stretch, lets try to mount the NFS server from an NFS client.

This example shows Linux as the client though any NFS-compatible client such as the various UNIXes, Windows 3.x/95/NT (with 3rd party software), etc. should work fine.

Mount the remote NFS share:

NOTE: Make sure that the client directory /mnt/nfs exists. If it doesn't, just do a "mkdir /mnt/nfs" first.

```
mount -t NFS 192.168.0.1:/home/hpe /mnt/nfs
```

- If all went well, the "mount" command should have executed quitely and returned you to the UNIX prompt. So go ahead and look around in the /mnt/nfs directory. You should see all of the remote files just as if they were local!

41 EXT2 File system tuning

[This is an on-going experiement but NONE of the following can hurt:]

Recently on a ~1500 user Linux box that I support, we have had major EXT2 filesystem corruptions on two seperate occasions. I then emailed several people about this and here are two replies I received:

From Warlock:

```
--
Personally, I have cron run ‘sync’ in the background every 10 minutes
or so and, averaged over any reasonable period of time, . . . (I have been
doing this) Forever.  . . . Doing a sync in the background every so often
(or between packages) pretty much fixed that problem.  Now everything is
much more stable, but the principle still holds.

I think the double-sync (old-timers use a triple, but our computers and
peripherals were slower back then) (:  is for when you want to *shut down*
(or reboot) and risk something very unclean.  Even if you type ‘sync’,
that isn’t guaranteed.  It basically tells the kernel to clean up and then
returns, but the actual process isn’t finished by the time sync finishes.
I think the logic was that a double-sync might block until the first
sync was finished, and a triple-sync was just there to but time for
the hard drive to finish writing out anything (disconnected SCSI drive,
for example).  I’m sure actually waiting 5-6 seconds after you typed the
first sync would be just as good 90% of the time, but you know humans.  (:
--
```

So, to implement this:

Redhat:

* edit /etc/crontab and append:

```
--
        0,10,20,30,40,50 * * * * root run-parts /etc/cron.10min
--
```

* Now create the dir /etc/cron.10min

```
--
        mkdir /etc/cron.10min
--
```

* create the simple file /etc/cron.10min/re-sync

```
--
        sync
--
```

* Make it executable:

```
--
        chmod 700 /etc/cron.10min/re-sync
--
```

* That's it. Cron will notice the changes and reload * automatically.

Slackware:

* edit /var/spool/cron/crontabs and append:

```
--
        0,10,20,30,40,50 * * * * root run-parts sync
--
```

* That's it. Cron will notice the changes and reload * automatically.

From the Yashy-Hack list:

```
--
Linux ext2 filesystems normally run asynchronously.  While this makes them
faster, it also makes them somewhat less reliable, especially on systems with
long uptimes.  If you're running a production machine (ie that people are
depending on), you can make filesystems run in synchronous mode by adding the
flag 'sync' to the options section in /etc/fstab - right now that section
likely says 'defaults', or maybe one of the quota options.  The filesystems
will be slower, but they'll also be more reliable.

<IMHO>

This is one reason I personally prefer FreeBSD for servers, though I use Linux
for my router and notebook, and frequently for workstations.  The BSD ufs
filesystem, which defaults to synchronous operations, is in my experience
more robust for long uptimes on heavily used systems.
```

```
>From the FreeBSD mount manpage:

     async   All I/O to the file system should be done asynchronously.
             This is a dangerous flag to set, and should not be used
             unless you are prepared to recreate the file system
             should your system crash.

</IMHO>
```

42 Dial-in terminal / PPP access via a modem

NOTE: There are several "gettys" out there and it isn't totally clear on how they are different. But, here is a little snipet from /usr/doc/getty_ps-2.0.7j/README.hi-speed:

```
--
I've only tested uugetty on dialin lines (with a Zoom v34X 36.6K) at
57.6 and 115.2Kbps.  I generally use agetty for dumb terminals,
mingetty for the console, and faxgetty calling agetty for combination
fax/data lines. (hylafax)
--
```

- edit /etc/inittab

Redhat: - Find the line that says: "6:2345:respawn.." and copy it to also say (for a modem on COM1):

```
"7:23456:respawn:/sbin/uugetty ttyS0 38400 vt100"
```

- Create the file /etc/default/uugetty.ttyS0 (for dial-ins on COM1)

NOTE: This config assumes you are using a modem on COM1, that it is going to answer the phone after -6- rings and before the user is shown a "Login:" prompt, the user will have to blindly enter in the password "letmein".

```
--
# [ put this file in /etc/default/uugetty.<line> ]
#
# sample uugetty configuration file for a Hayes compatible modem to allow
# incoming modem connections
#
# this config file sets up uugetty to answer with a WAITFOR string.  When
# using waitfor, it is necessary to specify INITLINE=cua?

# line to use to do initialization.  All INIT, OFF, and WAITFOR functions
# are handled on this line.  If this line is not specified, any other
# program that wants to share the line (like kermit, uucp, seyon) will
# fail.  This line will also be checked for lockfiles.
#
# format: <line> (without the /dev/)
INITLINE=ttyS0
```

```
# timeout to disconnect if idle
TIMEOUT=60

# modem initialization string: Sets the modem to disable auto-answer
#
# format: <expect> <send> ... (chat sequence)
#INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n AT\sM0\sE1\sQ0\sV1\sX4\sS0=0\r OK\r\n
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n ATS0=6\r OK\r\n

# waitfor string: if this sequence of characters is received over the line,
# a call is detected.
#WAITFOR=RING
WAITFOR=CONNECT

# this line is the connect chat sequence.  This chat sequence is performed
# after the WAITFOR string is found.  The \A character automatically sets
# the baud rate to the characters that are found, so if you get the message
# CONNECT 2400, the baud rate is set to 2400 baud.
#
# format: <expect> <send> ... (chat sequence)
#CONNECT="" ATA\r CONNECT\s\A
CONNECT=letmein

# this line sets the time to delay before sending the login banner
DELAY=1
--
```

- Finally, make sure your modem is connected and powered up and now tell Linux to initialize the modem with:

```
/sbin/init q
```

That's it. Go ahead, dial in with a modem and let it RING (6) times. After the sixth ring, the modem should answer and you should then be dropped to "nothing". Now blindly type in "letmein" and you should then see a normal Linux "login:" prompt.

42.1 For PPP connectivity:

To do your work via PPP instead of doing it via a standard terminal, follow the PPP setup recommendations in 22 (Section 22). Then, after you successfully login and are dropped to a UNIX prompt, simply type in the following (for a modem on COM1):

```
        /usr/sbin/pppd /dev/ttyS0 38400
```

NOTE: Many of you would probably rather have Linux default to a PPP only mode. To me, this is far more inflexiable and what happens if you aren't on a system that doesn't have PPP functionality? Doing it this terminal–>ppp way is MUCH more flexible.

42.2 Dialing in with answering machines:

- The following is VERY dependant on your home answering machine -

If you are like me, you only have one phone line and there is an answering machine on that line that answers the phone around call 3 or 4. To get past this, I can get into my answering machine remotely and turn it OFF. Once off, the linux's modem will answer after -6- rings. Once I'm done dialing in, I TEMPORARILY disable uugetty in /etc/inittab, rerun "/sbin/init q", and then re-call my answering machine with 15 rings. After that, the machine will turn back on. Once this is set, you'll need to re-enable uugetty in the /etc/inittab file and rerun "/sbin/init q" from a TELNET/SSH connection.

With that all behind you, if you ever make a mistake editing your IPFWADM rule sets, your Inet connection is down, etc, you now have a secured BACKDOOR into your machine!

43 Automated RPM notifiers

The tool "rpmwatch" creates reports based on Redhat's WWW site. As you might notice, this is only for Redhat and its RPMs. In addition to this, it does NOT work on Redhat's newer WWW pages nor sites for Mandrake, etc. Because of this, I have started implementing "AutoRPM" as shown below.

43.1 AutoRPM (the preferred solution):

- Download AutoRPM and the Perl "libnet" library from the URLs in 5 (Section 5)

- Uncompress AutoROM some temporary place like /usr/src/archive/rpm-tools/

```
tar xzvf autorpm-*.tar.gz /usr/src/archive/rpm-tools
```

- The LibNet module is a commonly installed tool with Perl. To verify that its already installed, run:

```
find /usr/lib/perl5/ | grep FTP.pm
```

if nothing shows up, LibNet isn't installed

- If it isn't installed, uncompress the LibNet library to a place like

```
/usr/src/archive/cpan

          tar xzvf libnet-*.tar.gz
```

- Next, got into the new libnet directory, compile, and install it:

```
cd /usr/src/archive/cpan/libnet-*
perl Makefile.PL
make
make test
make install
```

- Next, go into the new AutoRPM directory

```
cd /usr/src/archive/rpm-tools/autorpm-*
```

- Create its configuration directories

```
mkdir /etc/autorpm.d
mkdir /etc/autorpm.d/pools
```

- Copy over the program, the configuation files, and the man pages

```
cp autorpm.pl /usr/local/sbin
cp autorpm.conf /etc/autorpm.d
cp autorpm.d/* /etc/autorpm.d
cp pools/* /etc/autorpm.d/pools
cp autorpm.8 /usr/local/man/man8
cp autorpm.conf.5 /usr/local/man/man5
```

- Fix its permissions:

```
chmod 700 /etc/autorpm.d /etc/autorpm.d/pools
chmod 700 /usr/local/sbin/autorpm.pl
```

- Next, test it:

Mandrake 6.1 users:

```
/usr/local/sbin/autorpm.pl --ftp ftp.linux-mandrake.com:/pub/updates/6.1/RPMS/
```

Redhat 6.1 users:

```
/usr/local/sbin/autorpm.pl --ftp updates.redhat.com:/<url url="ftp:/6.1/i386/">
```

If that test works ok, time to tune your /etc/autorpm.d/setup:

Mandrake 6.1 users: ——————-

- Find the following lines in /etc/autorpm.d/autorpm.conf

```
/etc/autorpm.d/autorpm.conf
--
Config_File("/etc/autorpm.d/redhat-updates");
--
```

to

```
--
Config_File("/etc/autorpm.d/mandrake-updates");
--
```

- Create the file /etc/autorpm.d/pools/mandrake-updates . In this file, put at LEAST the following line on the top. If you want, you can add other Mandrake mirror URLs in this file as well. I have listed (2) others for an example:

```
/etc/autorpm.d/pools/mandrake-updates
--
ftp.linux-mandrake.com:/pub/updates/6.1/RPMS
rpmfind.net:/linux/Mandrake/updates/6.1/RPMS
ftp.orst.edu:/pub/packages/linux/mandrake/updates/6.1/RPMS
--
```

- Next, create the following file. Edit as you deem fit. Please note that I'm still in the process of learning and tuning this tool, if you have comments, etc, please let me know.

/etc/autorpm.d/mandrake-updates

```
--
##########################################################
# This one will mirror the updates for all versions
# of Red Hat 5.0, but won't bother with the source RPMs.
# All the updates stored locally will be in architecture-
# specific directories just like on the original site.

ftppool ("mandrake-updates") {

   # Recurse through the remote FTP site if necessary
   # Recursive (Yes);

   # Compare, recursively, the remote files to this directory
   # Recursive_Compare_To_Dir ("/usr/src/archive/md61-updates");

   # Ignore any directories named 'SRPMS' when recursing.
   # Regex_Dir_Ignore ("SRPMS");

   # What to do if the remote RPM is a newer version
   # that the local copy
   action (updated) {

      # Delete whatever local file we had that was older
      # than the remote file.
      # Delete_Old_Version (Yes);

      # Store the remote file in this local directory.
      # the 'Recursive' part means that if the remote
      # file was in the /i386/ subdirectory, it will be
      # stored in a /i386/ directory locally.
      # Recursive_Store ("/usr/src/archive/md61-updates");
          Install (Interactive);
          Report (Yes);
          Report_Queues_To ("root");
          Report_To ("root");
          Report_All (Yes);
            Display_Report (Yes);
   }
```

```
   # What to do if the remote RPM has no corresponding
   # version locally (e.g. it is new)
   action (new) {
          Install (Interactive);
          Report (Yes);
          Report_Queues_To ("root");
          Report_To ("root");
          Report_All (Yes);
          Display_Report (Yes);
    #  Store_Recursive ("/usr/src/archive/md61-updates");
   }
}
--
```

Once you are happy with how AutoRPM runs, I recommend have it run ONCE A DAY. To do this, do the following:

```
        ln -s /usr/local/sbin/autorpm.pl /etc/cron.daily/autorpm
```

Finally, I recommend to read the "autorpm" man page and pay attention to the "auto-ignore" file. There is a lot of other interesting info in the man page so I recommend that you read it. Its well written too!

43.2 rpmwatch

Download at RPM Watch from 5 (Section 5)

```
                rpm -Uvh rpmwatch-x.x-x.noarch.rpm
```

Create the file "run-rpmwatch" with the following contents:

NOTE: You need to edit the scripts to reflect your Redhat distribution installation. If you don't change the script to look to the proper URLs, your results will be worthless. On that same token, I request all the patches out there for ALL Redhat distributions though I only run 5.0. While this lets me know whats out there, some of the updated tools in 5.2 will NOT work correctly on 5.0 distributions. So, be careful and be SURE to read the "Testing RPMs before installing" at the top of 53 (Section 54) to see what files might be overwritten, etc.

/usr/local/sbin/run-rpmwatch

```
--
#!/bin/sh

# Version v1.2

echo "Getting RH50 errata.."
lynx -source <url url="http://www.redhat.com/corp/support/errata/rh50-errata-general.html"> > /tmp/rh50-errata-ge
lynx -source <url url="http://www.redhat.com/corp/support/errata/intel/rh50-errata-intel.html"> > /tmp/rh50-errat

echo "Getting RH51 errata.."
lynx -source <url url="http://www.redhat.com/corp/support/errata/rh51-errata-general.html"> > /tmp/rh51-errata-ge
lynx -source <url url="http://www.redhat.com/corp/support/errata/intel/rh51-errata-intel.html"> > /tmp/rh51-errat
```

```
echo "Getting RH52 errata.."
lynx -source <url url="http://www.redhat.com/corp/support/errata/rh52-errata-general.html"> > /tmp/rh52-errata-ge
lynx -source <url url="http://www.redhat.com/corp/support/errata/intel/rh52-errata-intel.html"> > /tmp/rh52-errat

echo "Converting to TXT..."
href2txt /tmp/rh5*-errata-*.html > /tmp/rh-errata.txt

rm -f /tmp/rh5*-errata*.html

echo "Running rpmwatch.."
rpmwatch -e /tmp/rh-errata.txt

echo -e "\n\nA good site to get all Errata RPMS is:"
echo "<url url="ftp://ftp.codemeta.com/pub/mirrors/redhat/updates/"">;

rm -f rh-errata.txt

echo -e "\nDone.."
--
```

- Now, make "run-rpmwatch" executable by running "chmod 700 rpm-watch"

- Run it by typing in "./run-rpmwatch"

The output should look something like:

```
[root@roadrunner tools]# ./run-rpmwatch
Getting RH50 errata..
Converting to TXT...
Running rpmwatch..

.       <skipping misc text>

FL RPM                              VERSION BUILD            UPDATE
--------------------------------------------------------------------
   samba                          1.9.18p10     5                ok
   rpm                                2.5.3   5.0                ok
   rpm-devel                          2.5.3   5.0                ok
 B bash                              1.14.7     6         1.14.7-11
```

*** NOTE: please see the bottom of this section on adding this script to a weekly CRON process!

* Regardless of the tool that you use, I'd recommend that you add it CRON to be executed once a week. Since RPMWATCH is the only tool currently running, I'll use that for an example:

Slackware:

Edit the file /var/spool/cron/crontabs/root and append the following:

```
--
# Run the sendlogs program at 12:00am everyday
02 2 * * 0 /usr/local/sbin/run-rpmwatch
```

Redhat users:

Create a symbolic link to point to the run-rpmwatch script:

```
ln -s /usr/local/sbin/run-rpmwatch /etc/cron.weekly
```

- That's it. Now, make cron re-read it's config files by doing:

- Redhat: killall -HUP syslogd
- Slackware: kill -HUP ‘ps aux | grep syslogd | grep -v -e grep | awk '{print $2}'‘

44 Nmap port scanner

Once you have secured your Linux box and implemented a good packet firewall, you need to TEST it to make sure you didn't miss anything. To do this, I recommend that you either port scan yourself from an unprivileged IP address or have a buddy do it for you.

The following instructions is on how to install Nmap and run it to check your host.

- Download the newest version of nmap from 5 (Section 5)

- Uncompress it (tar xzvf nmap-*.tgz)

- cd into the new nmap directory and run "./configure"

- Nmap will now configure itself

- Now just run "make" and then "make install"

- That's it! Nmap is installed! Now, nmap supports over 10 different port scans and running each one takes a while. So, I recommend that you setup this little script to ease the pain:

```
scan-it
--
#!/bin/sh

echo -e "\nPort Scanning $1 - TCP connect\n"
./nmap -sT $1
echo -e "\nPort Scanning $1 - SYN\n"
./nmap -sS $1
echo -e "\nPort Scanning $1 - FIN\n"
./nmap -sF $1
echo -e "\nPort Scanning $1 - Xmas\n"
./nmap -sX $1
echo -e "\nPort Scanning $1 - Null\n"
./nmap -sN $1
echo -e "\nPort Scanning $1 - UDP\n"
./nmap -sU $1
echo -e "\nPort Scanning $1 - Ident\n"
./nmap -I $1

echo -e "\n\n\nNmap done.\n\n"
--
```

- Next, make it executable by running "chmod 700 scan-it"

- Finally, to run a scan, just type in:

```
scan-it <ip>
```

Where <ip> is the IP address you want to scan. Once you start the scan, it will take a while so just relax and wait a while.

NOTE: Be warned:

- Nmap 2.0x port scans will CRASH Cisco IOS 11.3/x / 12.0.x routers that have SYSLOG enabled.

- If you implemented a IPCHAINS/IPFWADM rule set that logs failed connections, your logs will get MASSIVE. Many of NMAP's port scans scan all 65,535 ports. Now:

65,535 ports * 7 = 458,745 lines in your SYSLOG files!

45 So you think you are being hacked: Confirm it!

Once you've followed TrinityOS to a "T", you can be assured that your box is pretty stinken secure. BUT.. nothing is 100% secure and there will always be a chance that a hacker will find a way into your box.

With this in mind, please read what Brad Alexander had to say:

```
"As with system administrators and security specialists, there are
varying levels of skill among the system crackers. The notes included
in this document, and in fact, any notes about what to look for is
subjective, since the cracker will endeavor to cover his tracks. This
may include the use of a rootkit, which inserts trojaned binaries such
as "ls", "login", "ps" and so forth and hides sniffers on your system,
editing out parts of your logfiles, and the like. The attacker may
create directories such as "..." or ".. " to hide his warez. The attack,
like the individual cracker, will have different personalities. Your best
bet, aside from keeping the intruder out, is to run overlapping layers of
intrusion detection software, both host-level (such as Abacus Sentry) and
network level (such as SHADOW and Network Flight Recorder). If the cracker
attempts to disable one system, it will trigger another. The same should
be said for your file monitors, (e.g. Tripwire and ViperDB). However, there
is no substitute for a familiarity with your system and your filesystem."
```

Couldn't have said it better. So, with all that in mind, here is my best initial stab at figuring out if you've been hacked:

Here is a quick list that you can follow:

1) Check for any "ESTABLISHED" connections to your box by running "netstat -a | more". If there are connections to your box other than SMTP (port 25 for mail), DNS (port 53), and possibly WWW (port 80) that you don't know about, this should raise a flag. Especially look for SSH, TELNET, or FTP conenctions.

2) Using your favorite file viewer (vi, Pico, less, etc), look at your log files for strange things like:

- changed passwords

- strange connections from unknown IPs

You can also use the "pwck" and "grpck" commands to check these file too.

3) Run "last | more" command to see what users have recently logged into your machine.

4) Check the date of the /etc/shadow file to make sure it hasn't been recently changed

5) If you question the integrity of any of your executable files, verify that they are ok:

Redhat:

```
rpm -Va
```

or you can use the following script:

```
--
#!/bin/sh

for pkg in ‘rpm -qa‘; do
  echo "Verifying $pkg" >> /tmp/verify.log
  rpm --verify $pkg >> /tmp/verify.log
done
--
```

If your box HAS been compromised:

1) Disconnect the machine's network connection, be it a modem, Ethernet connection, etc.

2) Try to determine what the hacker did to your box:

- look at /root/.bash-history
- look at the slackware:/var/adm or redhat:/var/log log files

3) If you installed Tripwire, re-run it and see what files were changed.

If your machine was compromised and you are unable to determine what was hacked, you have to consider that ALL security on this box has been breeched. Because of this, you'll need to backup all changed user files (NO EXECUTABLE FILES WHAT SO EVER), wipe ALL HDs and either restore from a known good backup or re-install the OS from scratch! Ouch!

[Once I get more time, I will expand on this section]

46 UNIX and Samba Printing

This example is primarily to get Samba printing working but it will work fine for local UNIX printing too. This example assumes you have a HP LaserJet IIp and its connected on LPT1 (not LPT0).

- It has been usually understood that using the BSD "lpd" program is a *HIGH* security risk. The reason for this was because the various "lp" tools have SUID ROOT permissions. Meaning that when anybody runs the "lpr" program, the program will actually run as if "root" ran it.

Though we can't do anything about this for "lpr", we can fix things for "lpd" Increase the permissions on the /dev/lp* devices and remove the SUID bit from "lpd". What does open up the permissions on /dev/lp* do against you? People could possibly cat text to it and make it run out of paper but who cares!!!

The permissions were in /usr/bin/

```
--
-r-sr-sr-x   1 root     root        13876 Oct  1 21:55 lpq
-rwxr-xr-x   1 root     root         2406 Aug 15  1998 lpqall.faces
-r-sr-sr-x   1 root     root        15068 Oct  1 21:55 lpr
-r-sr-sr-x   1 root     root        14732 Oct  1 21:55 lprm
-rwxr-xr-x   1 root     root         3492 Oct  1 21:55 lptest
-rwxr-xr-x   1 root     root         2507 Oct 11 00:15 lpunlock
--
```

to

```
chmod 700 /usr/sbin/lpd
chmod 755 /usr/bin/lp*
chmod 4755 /usr/bin/lpr
```

and

```
chmod 660 /dev/lp0
```

One note about the file permissions on "lpr" from 8 (Section 8)

```
#NOTE: I feel setting "lpr" to allow any group to execute it is
#        a bad thing.
#
#        I would like to add UNIX users and even the Samba process to
#        the "lp" group already defined in /etc/groups and then be able
#        to put things back to to 4750.  BUT.. I just talked to a buddy
#        of mine and this really isn't possible.  Linux doesn't support
#        multiple groups per file and Linux doesn't support access lists
#        (ACLs') yet.  So, you either have to do all this or run LPRng.
```

- Next, create the /etc/printcap file and put in the following. Please note that this example is for a HP LaserJet IIp on LPT1 and a Epson Stylus 500 Color ink jet on LPT2.

The following "lp" setting is for local UNIX printing and "Hp_Lj2p" is for Samba printing

```
      --
      ##PRINTTOOL3## LOCAL ljet2p 300x300 letter {} LaserJet2p Default 1
      lp:\
      :sd=/var/spool/lpd/lp:\
      :mx#0:\
:sh:\
      :lp=/dev/lp1:\
      :if=/var/spool/lpd/lp/filter:

      ##PRINTTOOL3## LOCAL epsonc 240x216 letter {} EpsonLQ24 Default {}
      lp2:\
      :sd=/var/spool/lpd/lp2:\
      :mx#0:\
      :sh:\
      :lp=/dev/lp2:\
```

```
        :if=/var/spool/lpd/lp2/filter:

Hp_Lj2p|raw:\
        :rw:sh:\
        :mx#0:\
        :lp=/dev/lp1:\
        :sd=/var/spool/samba:\
        :fx=flp

Epson_S|raw:\
        :rw:\
        :sh:\
        :mx#0:\
        :lp=/dev/lp2:\
        :sd=/var/spool/samba:\
        :fx=flp
--
```

- Next, you need to re-enable "lpd" from 8 (Section 8) and then load up the lpd daemon:

- Redhat: /etc/rc.d/init.d/lpd start

- Slackware: /usr/sbin/lpd -l&

- If you are runngin Samba, you'll have to edit your /etc/smb.conf file as shown in the Samba section of TrinityOS and then re-start the SMB process.

- From here, Samba Printing should work fine.

- If you want to do native UNIX printing, it starts to get VERY crazy without a configuration tool. I could post my /var/spool/lpd/lp/filter file but its over 9K and specific to the way Redhat does things! So, I highly recommend to a GUI tool native for your specific distribution.

- Redhat:

Xwindows-GUI: printtool (via control-panel)

NOTE: The Hp Laserjet needs the "anti-staircase" option

- Slackware: ???

- Once the GUI tool sets up your printer, things should be good to go. To be honest, it SUCKS that I'm not documenting how to do it via a command line but I have to say that UNIX printing is so damn hard! Oh well.. sorry!

47 IPSec (SWAN) Virtual Private Network (VPN) [Almost complete]

IPSEC is the new, standards-based way of setting up a Virtual Private Network (VPN) between two computers. Though IPSEC was originally designed for the new IPv6 (IPng) TCP/IP protocol, it is also being deployed for the TCP/IPv4 (normal TCP/IP) too.

If you don't know what a VPN is, imagine a network at work that is on the Internet but behind a strong firewall. Unless you have remote access into work, you can't get to any of those machines huh? Not anymore! If your work has a connection to the Internet and a IPSEC VPN server (be it Linux, Cisco, etc), you'll now have ability of accessing your computers internal to your work via the Internet in a secure and 168-bit+

encrypted fashion. Though you're access speeds and even availably will be Internet-weather dependant, its both a GREAT and CHEAP method of remote access.

Common questions include:

* Is IPSEC only for Linux? No way! Who else can connect?

Currerntly, there have been several ports that Linux's SWAN IPSEC VPN works with:

- YES: Cisco IOS-based routers (in 168bit 3DES mode not 1DES)
- YES: Axent's Raptor VPN
- NO: Bay Contivity Extranet v2.02 in either "Single Client - Agressive" mode or "Remote Network - Main mode"

I'm sure other vendors will be added to this list as time goes on.

* Is it RFC complient?

Linux FreeS/WAN is an implementation of IPSEC. It does not yet implement all of IPSEC, but everything it does follows the IPSEC RFCs.

* What about Performance and CPU utilization?

Someone has tested the SWAN VPN with a Cisco 2501 and a 486/DX50 across as T1. The 486's CPU utilization was about 15% while the 2501's utilization was about 80%!

One benchmark seens with Triple DES (our default bulk encryption method) can do 1.6 megabytes per second on a Pentium 200. That's > 10 megabits/second.

(on a 100Mbit LAN: with the OLD SWAN code : Newer SWAN code should run roughly 3x faster on Intel x86 systems:

```
                        * No IPSec             * DES                              * 3DES
                        P200 = >80 Mb/s P200 = 10-15 Mb/s              P200 = 2-4 Mb/s
                        P450 = >80 Mb/s P450 = 20-25 Mb/s              P450 = 10-14 Mb/s
```

I think encryption is what degrades performance the most, and you would be best off with a HW accelerator if you want to get closer to max.

*** NOTES:

- Please note that I haven't had the time to bring this up myself yet but I've had a few users that said that they did. If you have any comments, ideas, changes, please email me.

- Please see the Gotchas at the end of this section regarding DHCP, IPCHAINS/IPFWADM rule sets, etc.

- If you have problems with the SWAN code, please join the SWAN email list for support. I cannot help at the moment since I don't have a SWAN setup running

```
--
FreeSwan/IPSec installation instructions for Linux

v1.20           Clarifications made and added a Gotcha regarding : dranch
v1.10   Additions by David A. Ranch
v1.00   by Rob Hutton   <mailto://HuttonR@plymart.com>
```

NOTE: You should also be able to terminate the VPN on the Linux box directly. This isn't documented here yet but it will be done in the TrinityOS doc. Until then, you'll have to figure it out.

NOTE2: This document assumes that you are running this initially WITHOUT a firewall. Once its running, see the bottom for the relivant IP ports to allow though the IPFWADM/IPCHAINS/etc rule sets.

If you have not configured and built your own kernel, do so. The FreeSwan utilities depend on the results. Instructions for that can be found at

`<http://www.tldp.org/HOWTO/Kernel-HOWTO.html>`

Once you have compiled and built your own kernel, draw a simple diagram as follows:

```
|   Machine (S)  |      |     Machine (G)   |      |     Machine (H)   |      |   Machine (T)  |
|   Remote Host  |<--->|Remote Firewall/VPN|<...>|Local Firewall/VPN|<--->|   Local Host   |
|   IP:          |      | IP:               |      | IP:               |      |   IP:          |
```

Record all IP addresses, and their associated interface and netmask, and the routing tables from each machine. Then, it is CRITICAL to first TEST you network connectivity before you attemp to setup the VPN. It is recommended that the (S) machine can ping (T) and that (T) can ping machine (S). Also test any other services that you will be using such as TELNET, SSH, FTP, SMTP, etc .

NOTE: If *either* protected network is privately addressed, please see the note in the "Notes and Gotchas" Section.

[DO THE FOLLOWING ON BOTH MACHINES]

Download the newest version of SWAN (preferably the current "snapshot" code) from the sites found in 5 (Section 5)

Uncompress the file using:

```
        tar xvzf freeswan-X.tar.gz
```

or your favorite uncompress command where "X" is the newest version of SWAN.

This will create a directory called freeswan-X with the sources and installation files in it. I recommend that you print the INSTALL and doc/vpn.how file to refer to.

cd to the freeswan-X directory. Build the libraries, programs, and utilities by typing:

```
        make
```

Then install them by typing:

```
        make install
```

Edit the /etc/sysconfig/ipsec file. Look for the KLIPSINTERFACES variable. Change it to reflect the interface that you will be using to run the VPN across.

NOTE: This assumes you are running Redhat Linux

Next, install the kernel patches be typing:

```
        make insert
```

CD to the LINUX source directory and run menuconfig:

```
cd/usr/src/kernel/linux
make menuconfig
```

The following networking options should now be set on:

- IP: forwarding/gatewaying
- IP: tunneling
- Kernel/User network link driver

If it is not enabled, set the following on:

- IP: optimize as router not host

You should also have new options at the bottom of the page for "IP Security Protocol (IPSEC)" which should be enabled. Now exit and save your configuration, and remake and install the new kernel. When you are finished, reboot to activate the changes.

Next, edit the /etc/services file and add the following (if not there already):

```
--
isakmp          500/tcp    isakmp
isakmp          500/udp    isakmp
--
```

Again, verify that you can ping, telnet, ftp, etc. from one host/workstation to the other (T to S and S to T) in both directions.

[DO THE FOLLOWING ON ONE OF THE FIREWALLS. I WILL USE G]

Edit the /etc/ipsec-auto file. Change the left=[id address] to be the ip address of the NIC you are running the VPN across on machine G. Change leftsubnet=[ip address/netmask bits] to the address/netmask of the private/protected subnet on machine G. If the machines are not directly connected (on the same network), change the address of leftnexthop=[ip address] to the address of the next router between G and H. Now edit the corresponding "right" variables to match the configuration of H. Exit and save your changes.

Edit the /etc/ipsec-manual file. Make the same changes to the snt connection and delete all of the other connections. Exit and save your changes

Edit the /etc/isakmp-secrets file. Change the IP addresses (the first column) to match the addresses of the nics that are running the VPN. Exit and save your changes.

Copy the ipsec-auto, ipsec-manual, and isakmp-secrets from G to H. Using a floppy is the best way to make sure that the files do not get corrupted. Make sure that the files on both machines are owned by root and have permissions rw——-.

Again, reboot both machines.

Examine the /var/log/messages (for Redhat users) to make sure that IPSEC loaded without any error messages. Also, verify that the following entries exist in the /proc/net/ directory:

```
ipsec_eroute
ipsec_spi
ipsec_spigrp
ipsec_spinew
ipsec_tncfg
ipsec_version
```

Verify that ipsec is attached to the correct NIC by typing:

```
        cat /proc/net/ipsec_tncfg
```

on (host G) type:

```
        ipsec manual snt up
```

Then on (Host H) type the same thing.

Now type ipsec look on either machine. You output should look something like:

```
foo.spsystems.net Wed Nov 25 22:52:45 EST 1998
-------------------------------------
10.0.1.0/24 -> 11.0.1.0/24 => tun0x200@11.0.0.1 esp0x2@11.0.0.1
-------------------------------------
tun0x200@11.0.0.1 Ipv4_Encapsulation: dir=out     10.0.0.1 - > 11.0.0.1
etc.
etc.
etc.
```

If it does, your VPN is up. You can test it by doing a tcpdump in between the two machines. You should see data transmitted back and forth over IP protocol 50. Test each subsystem to make sure they work using FTP, TELNET, SMTP, etc.

Now type the following on both boxes:

```
        ipsec manual snt down
```

Now type the following on both boxes:

```
        ipsec auto snt add
        ipsec auto snt up
```

Again, test to see that each subsystem works.

Auto-starting the VPN:

Edit the /etc/sysconfig/ipsec file on both machines. Near the bottom add "snt" to both the PLUTOLOAD and PLUTOSTART variables. Now reboot both machines, and the VPN should start automaticly.

47.1 Bugs and Gotchas:

47.1.1 Newest fixes and patches:

The latest SWAN code is always in the snapshot.tar.gz file. If you cannot get SWAN to work, etc, you might want to try installing the snapshot as there have been many changes since the x.91 code was released.

47.1.2 Private addressing:

If either network is privately addressed and you are running over the internet you will not be able to do this. In this case, if you can ping devices on the internet outside of your network from the VPN servers (machines G and H), routing is probably correct. Once the tunnel is up, you will not be able to see any machine on the remote subnet from the gateway machine (G or H), so make sure you are testing the VPN from client machines on the protected subnets, not the gateway machines themselves.

47.1.3 DHCP

Currently, DHCP will return with an unknown device type error after you install the SWAN patches (It will do this whenever you set up a tunneled interface) and then exits. To fix this, download the DHCP source from the URL in 5 (Section 5). Next, in the DHCP source code, ADD the following BEFORE the "ARPHRD_ETHER" case statement:

NOTE: This issue might have been fixed in newer released of Swan

```
common/dispatch.c
--
        case ARPHRD_TUNNEL:
      /* ignore tunnel interface */
    break;
--
```

After this done, compile DHCP per the instructions in the README

47.1.4 Automatic SWAN startup

The other problem is that the automatic startup documented above does not work. They are looking at why now. There is a workaround. It is as follows:

Create a rc.ipsec in the rc.d directory. For each connection add the following to it:

```
ipsec auto [connection name] add
ipsec auto [connection name] up
```

...[eof]

Set the file permissions to rwx——-. Then run it from the rc.local

47.1.5 Running SWAN through a IPFWADM/IPCHAINS/other firewall:

You have to allow the IPSEC traffic through your IPFWADM/IPCHAINS firewall rule sets. Port 500 is the key negotiation daemon. The ISAKMP tool does the key negotiation and then passes the keys to the daemon that runs the VPN. In FreeSwan, the daemons are called Klips and Pluto respectively.

Once you run the "ipsec auto [connection name] add" there is an interface called ipsec0, ipsec1, etc.

According to the programmer, port 92 is used in both directions, but when I set up my rules this way, I cannot get the tunnel up, so I'm going to do some more packet captures. After further investigation, I found the following rules to work:

NOTE: "other end's IP" is the remote VPN machine' Internet (external) IP address "this end's IP" is the local VPN machine's Internet (external) IP address

IPFWADM 2.0.x kernels:

```
--
## Inbound Ruleset
/sbin/ipfwadm -I -a accept -b -W $EXTIF -P udp -S [other end's IP] isakmp -D $EXTIP isakmp

## Outbound rule set
/sbin/ipfwadm -O -a accept -b -W $EXTIF -P udp -S $EXTIP isakmp -D [other end's IP] isakmp
--
```

IPCHAINS 2.2.x kernels:

```
--
## Inbound Ruleset
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p udp -s [other end's IP] isakmp -d $EXTIP isakmp

## Outbound Ruleset
/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p udp -s $EXTIP isakmp -d [other end's IP] isakmp
--
```

48 PPTP support as a Linux client or PPTP through a MASQ server

This section helps the reader to set up a Linux machine to be a PPTP client. This section also details how to enable an IPMASQ server and/or a strong IPCHAINed firewall server (NO solution is available for 2.4.x kernels with IPTABLES) to properly forward PPTP traffic.

Most typical Windows VPN clients that FORCE all network traffic through the VPN connection if it's up and running. Linux VPNs don't require this. This flexibility lets you keep your personal traffic on your own Internet connection while work traffic only goes over the VPN connection. Some Corporate IT people consider this a security issue and it CAN BE if your PPTP client machine is not secured. Yet, if you have properly followed most aspects of TrinityOS, you are pretty secure. :-)

Currently, this section makes TWO assumptions:

- The PPTP client is NOT the MASQ server. This isn't recommended since if your MASQ/Firewall machine is broken into, the hostile user will have full network connectivity over the VPN as well. If this doesn't concern you, you will have to make additional changes to the FORWARDING section to properly NAT this local traffic. Currently, this section does not cover this.
- ALL Corporate VPN traffic to/from the RFC1918 172.16-31.0.0/12 network address space will go over the PPTP VPN. All other traffic will go out your normal IP network connection.

48.1 Kernel source tree

When compiling up the various PPTP client software, you will NEED to have the kernel source code installed on your machine, configured via "make config", and the "make dep" process completed. There isn't any direct (need to actually compile the kernel or any kernel modules. In my case, I have the generic Linus 2.4.18 kernel installed with the sources saved under the "/usr/src/kernel/linux" path.

NOTE:

You SHOULDN'T need to recompile your kernel unless:

- PPP support is directly compiled PPP into the kernel (TrinityOS usually recommends this method). Please see below to learn how to tell if PPP is a module or not.
- If the PPTP client will be behind a IPMASQ server, the MASQ server's kernel will need to be recompiled to support PPTP/GRE. More details are provided later in this section.

48.2 Install PPTP related software

Download both the "pptpclient" client and the "ppp-mppe" PPP shim software from the URLs found in 5 (Section 5). I recommend to save these files in the "/usr/src/archive/pptp/" directory.

48.2.1 Confirm that your kernel is PPTP compatible

Before you start, it is critical to know that your kernel supports PPP via kernel modules and NOT via being built monolithically into the kernel. To verify this, you should get a directory listing of the following directory for your given kernel version:

For a 2.4.18 kernel:

```
# ls /lib/modules/2.4.18/kernel/drivers/net/

bsd_comp.o  ppp_async.o    ppp_generic.o  slhc.o
dummy.o     ppp_deflate.o  ppp_synctty.o
```

You need to have the "ppp_async" and "ppp_generic" modules listed. If you don't have these options, you'll need to recompile the kernel with modularized PPP options. For more information, please see 12 (Section 12).

48.2.2 Install ppp-mppe

MPPE stands for Microsoft Point-to-Point Encryption which we need to add to PPPd. This and a few other kernel modules makes Linux interoperate with Microsoft's 40-bit and 128-bit PPTP servers.

As you'll see, this package comes with PPPd 2.4.0, which is already outdated since 2.4.1 has been released. In my case, my Linux distribution comes with PPPd version 2.4.1 so installing MPPE downgraded it to 2.4.0. I didn't really care as 2.4.0 seems to work just fine. In the future I supposed they will release a ppp-mppe version with PPPd 2.4.1.

To compile and install the various PPTP software, first be sure you are ROOT. Then run these commands:

```
cd /usr/src/archive/im
tar xzvf ppp-mppe-2.4.0-4.tar.gz
cd ppp-mppe-2.4.0-4

. unpack.sh

cd ppp-2.4.0

./configure
make
```

Now let's save original PPP programs from your Linux distribution (please note that your distribution might place these files in different directories).

```
cp /usr/sbin/chat /usr/sbin/chat.bak
cp /usr/sbin/pppd /usr/sbin/pppd.bak
cp /usr/sbin/pppdump /usr/sbin/pppdump.bak
cp /usr/sbin/pppstats /usr/sbin/pppstats.bak
```

Now, let's install the new versions of PPPd:

```
make install
cd linux-kernel
```

NOTE: When compiling the MPPE kernel module:

- For some reason, the makefile's auto kernel-tree mechanism was broken for my machine. To fix things, I had to do the following (these steps assume that your linux source tree is in "/usr/src/kernel/linux"):

```
      edit the "kmodbuild.sh" script and find the "ARGS" line
      change it to read:

           ARGS="TREE=/usr/src/kernel/linux"
```

Ok.. now compile the kernel module:

```
./kmodbuild.sh
```

The final compile output from the above step should look something like:

```
There is a script in kernel-modules that can do this for you. To use it to
install your newly built kernel modules, type:

        kernel-modules/kmodinst.sh kernel-modules/new-2.4.18
```

Check the bottom line displayed on your system when you ran the "./kmodbuild.sh" script. The name of the directory will be different from the one displayed below depending on the kernel version installed on your machine.

From the message received from above, run the following command for a generic 2.4.18 kernel:

```
kernel-modules/kmodinst.sh kernel-modules/new-2.4.18
```

NOTE:

- Under the 2.4.19-pre5 beta kernel, I received dependency errors in the ppp_generic.o module (out_of_line_bug). Once I used a stock 2.4.18 kernel, things worked fine.

Now finished with MPPE section, lets get into the PPTPCLIENT installation

```
cd ../..
```

48.2.3 Install pptpclient

The PPTP client software is actually a VERY complex Perl script. Though I suppose I could have wrote something simple up on my own, this tool works just fine and offers some advanced features some users might like.

To install it, do the following commands:

```
tar xzvf pptp-linux-1.1.0-1.tar.gz
cd pptp-linux-1.1.0-1

#Yes, this is weird to have a tar in a tar but that's how the archive comes
#
tar xvzf pptp-linux-1.1.0.tar.gz
cd pptp-linux-1.1.0

make

cp pptp /usr/sbin
```

Finishing up:

```
cd ..
cp pptp-command /usr/sbin
```

NOTE:

Some users had to edit this "pptp-command" Perl script file and remove the "-T" option at the top of the Perl script file (I didn't):

```
    old: #!/usr/bin/perl -wT
    new: #!/usr/bin/perl -w
```

48.3 Create the various PPP/PPTP configuration files

Ok, from the PPTP archive, copy over the example OPTIONS file:

```
cp options.pptp /etc/ppp
```

48.3.1 Create the PPP peer file

The above installed "ptp-command" Perl script can be run without any command line arguments and run as an interactive program. Instead, I recommend to simply create the following files and edit them when required to match your setup.

So, copy the following text and save it as the file "/etc/ppp/peers"

NOTE:

- The "#" lines at the top of the "/etc/ppp/peers"file are NOT COMMENTS! They are actually parsed out by the "pptp-command" script. So if you wish to change the remote PPTP server's IP address or some of the routing commands, edit these specific #'ed out lines.

- EDIT: Please change the "YourUserNameHERE" and "REMOTE-PPTP-CHAP-HERE text to reflect your PPTP login name and the remote PPTP server's CHAP secret.

/etc/ppp/peers

```
#
# PPTP Tunnel configuration for tunnel MyEmployer
# Server IP: 220.1.2.3
# Route: add -net 172.16.0.0 netmask 255.240.0.0 dev TUNNEL_DEV
#

#
# Tags for CHAP secret selection
#
name YourUserNameHERE
remotename REMOTE-PPTP-CHAP-HERE

#
# Include the main PPTP configuration file
#
file /etc/ppp/options.pptp
```

Now, make this new file the default PPPd peers file:

```
ln -s /etc/ppp/peers/MyEmployer /etc/ppp/peers/__default
```

48.3.2 Create the chap-secrets file

Now edit the CHAP secrets file and put in your PPTP username and password.

VERY IMPORTANT NOTE:

Currently, your PPTP password will be saved in CLEARTEXT which is VERY BAD. I plan on updating this section to prompt for your password and NOT store it anywhere. Until then, just be sure that you fix the permissions of this file as shown below.

Please change the:

- "YourUserNameHERE" word to reflect your remote PPTP server's CHAP secret
- "PPTP-passwd" word to reflect your PPTP password

/etc/ppp/chap-secrets

```
# Secrets for authentication using CHAP
#
# client          server                     secret         IP addresses
#
YourUserNameHERE  REMOTE-PPTP-CHAP-HERE      'PPTP-Passwd'
```

IMPORTANT:

As mentioned above, be sure to only allow the ROOT user to be able access this file as your PPTP password is stored in there.

```
chmod 600 /etc/ppp/chap-secrets
```

48.3.3 Create the resolv.conf file

When the PPTP VPN connection is up, you need to make sure you use the DNS servers on the other side of the VPN so you can reach the intended private systems. Without this, nothing would resolve and thus, you wouldn't be able to connect to any internal machines by NAME though by IP would work.

NOTE:

- Currently, this section uses a static DNS server setup. It should be mentioned that PPPd has the ability to dynamically create the /etc/resolv.conf file. This section will be updated to reflect how to use a dynamically created /etc/resolv.conf file. Until then, the following steps will work fine.

Save your original "/etc/resolv.conf" as "/etc/resolv.conf.real"

```
cp /etc/resolv.conf /etc/resolv.conf.real
```

Next, create a "/etc/resolv.conf.pptp" file from the example text below. Please change the IP addresses here to reflect the correct INTRANET DNS servers that are on the other side of your VPN connection (myemployer.com).

/etc/resolv.conf.pptp

```
search MyEmployer.com
nameserver 172.24.244.10
nameserver 172.24.245.10
```

As a heads up, when you run "pptp-command start" script, the script will make a backup of your /etc/resolv.conf file and then copy the "/etc/resolv.conf.pptp" file over it. When you disconnect from the PPTP VPN with the "pptp-command stop" command, the script will copy the backup "resolv.conf.real" file back to to "resolv.conf".

48.4 Running PPTP for the first time

The first time you run the "pptp-command" script, I recommend to activate PPP's "debug" option. To do this, add the following line at the of beginning of the "/etc/ppp/options.pptp" file:

```
debug
```

In a different terminal/xterm, run the "logit" script from 9 (Section 9) to see what happens in real-time.

48.4.1 Load the PPP/PPTP kernel modules

Your system might or might not automatically install the following kernel modules automatically. Try running "pptp-command start" as show below and see if things work. If not, try the following:

```
/sbin/modprobe mppe

/sbin/modprobe ppp_async
```

After you do this, make sure that the following kernelmodules are loaded by running the "/sbin/lsmod" command. Please note that ALL of these modules are CRITICAL even if this isn't over a modem connection, etc. Trust me!

```
mppe                  20416   0  (unused)
ppp_async              6128   0  (unused)
ppp_generic           15088   0  [mppe ppp_async]
slhc                   4272   0  [ppp_generic]
```

48.4.2 Start up the PPTP VPN

Ok.. try it out:

```
pptp-command start
```

The script will start in background after a while... don't forget to check your log file to see what happens optionally using the LOGIT script.

48.4.3 Stop up the PPTP tunnel

To shut down the tunnel, run the following command:

```
pptp-command stop
```

48.4.4 Cleaning up

Once you are sure the PPTP setup is working, be sure to REMOVE that "debug" option mentioned above.

48.5 Running PPTP behind a Linux IPMASQ NAT or Strong firewall server

If you are running a strong IPCHAINS ruleset for firewalling or IPMASQ and firewalling (TrinityOS firewall, etc.), you need to add the following firewall commands to your rc.firewall ruleset to let the PPTP and GRE traffic through:

An example of a IPCHAINS firewall (not MASQing):

```
#portions of this ruleset are from TrinityOS(tm)

#pptp.Myemployer.com
SECUREHOST="220.1.2.3"

# -- INPUT SECTION --
#
# For just a strong firewall on the PPTP client itself
#
echo "     * Allowing $SECUREHOST INPUT for PPTP, GRE"
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST 1723 -d $EXTIP
```

```
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p 47 -s $SECUREHOST -d $EXTIP

# -- OUTPUT SECTION --
#
echo "     * Allowing $SECUREHOST OUTPUT for PPTP and GRE"
/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP -d $SECUREHOST 1723
/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p 47 -s $EXTIP -d $SECUREHOST
```

An example of a IPCHAINS Firewall with MASQing:

```
#portions of this ruleset are from TrinityOS(tm)

#pptp.myemployer.com
SECUREHOST="220.1.2.3"

#your EXTERNAL IP address  -- change this to be your PPTP client's IP address
#PPTPCLIENT=$EXTIP
PPTPCLIENT="1.2.3.4"

# -- INPUT SECTION --
#
# For just a strong firewall on the PPTP client itself
#
echo "     * Allowing $SECUREHOST INPUT for PPTP, GRE"
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p tcp -s $SECUREHOST 1723 -d $EXTIP
/sbin/ipchains -A input -j ACCEPT -i $EXTIF -p 47 -s $SECUREHOST -d $EXTIP

# -- OUTPUT SECTION --
#
echo "     * Allowing $SECUREHOST OUTPUT for PPTP and GRE"
/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p tcp -s $EXTIP -d $SECUREHOST 1723
/sbin/ipchains -A output -j ACCEPT -i $EXTIF -p 47 -s $EXTIP -d $SECUREHOST

# -- FORWARD SECTION --
#
/sbin/ipchains -A forward -j MASQ -i $EXTIF -p 47 -s $PPTPCLIENT -d $SECUREHOST
```

48.6 Troubleshooting your PPTP connection

48.6.1 PPTP through a IPMASQ server

If you are running a Linux / Windows /etc. PPTP client BEHIND an Linux IPMASQ server, you will have to apply, recompile, and reboot the MASQ server's kernel with the PPTP MASQ kernel patches. These patches allow Linux to:

- properly MASQ multiple PPTP clients to one remote PPTP server
- properly handle GRE packets used by PPTP VPNs

Please see the PPTP VPN URL in 5 (Section 5) to get the required patches for your kernel. Once the kernel has been patched, you will then have to configure the kernel with the following "Network" option:

```
IP: PPTP masq support (CONFIG_IP_MASQUERADE_PPTP) [Y/m/n/?] Y
```

NOTE:

- As mentioned at the top of this section, there currently isn't a PPTP solution for IPTABLES and 2.4.x based kernels.

If you get stuck on item #8 from the Advanced Troubleshooting PPTP URL from 5 (Section 5), try this:

```
modprobe ppp_generic
modprobe mppe
```

Final NOTE (whew!):

- If you see that when you get the PPTP tunnel running but your machine then starts to firewall traffic that shouldn't be firewalled (according to the SYSLOG logs), make sure that the file "/etc/ppp/ip-up.local" file doesn't exist.

49 IDE HDs performance optimization via hdparm

With the invention of IDE hard drives, which replaced the classic MFM, RLL, and even ESDI HDs of the past, things got much easier and cheaper. Unlike IDE, SCSI usually operates at top performance where as IDE must be tuned for the system the IDE HD is installed into. The command to do this with Linux is HDPARM.

With IDE HDs, you can configure (read the hdparm man page for a full list of feature with better descriptions):

- multicount - The number of sectors the HD can transfer per system interupt. Default is 1 and the max depends on your HD.

- I/O support - The data transfer mode the HD operates in. The default is 16bit though you can put it into 32bit mode if the IDE controller supports it.

- unmaskirq - This allows the OS to listen to other interrupts while a data transfer is taking place. Though this can speed things up, this can make your system unstable if you have a poor IDE chipset. Read the man page for more details.

- using_dma - With new UltraDMA (UDMA) hard drives and supporting UDMA controllers. UDMA is a technique to let the IDE chipset transfer data directly from HD to memory without bothering the CPU. Because of this, you can greatly reduce CPU utilization for big IDE transfers.

NOTE: I've tried using this parameter in the past and it ALWAYS has crashed the machine (P-II 400Mhz with IBM 16.8GB UDMA HDs). Your milage will vary.

Anyway, first, lets get an idea of what HDPARM see's for /dev/hda (my first IDE HD): Notice that I use "-I" to get the current HD setup settings:

```
                /sbin/hdparm -I /dev/hda

                /dev/hda:
                 Model=DW CCA1300H0, FwRev=911.E922, SerialNo=DWW-2T27
                 Config={ HardSect NotMFM HdSw>15uSec SpinMotCtl Fixed DTR>5Mbs FmtGapReq }
```

```
                 RawCHS=2100/16/63, TrkSize=57600, SectSize=600, ECCbytes=4
                 BuffType=3(DualPortCache), BuffSize=128kB, MaxMultSect=16, MultSect=16
                 DblWordIO=no, maxPIO=1(medium), DMA=yes, maxDMA=2(fast)
                 CurCHS=2100/16/63, CurSects=2116800, LBA=yes, LBAsects=2116800
                 tDMA={min:150,rec:150}, DMA modes: sword0 mword0 mword1
                 IORDY=on/off, tPIO={min:380,w/IORDY:180}, PIO modes: mode3
```

What does all this mean? Ok:

Line 1 - Its a Western Digitial drive with a model# of CCA1300H0 with the serial# (why is WD reversed? Dunno.. it doesn't that with the "/sbin/hdparm -i" command. Eh...)

- Line 2 - This lists the HDs technical abilities. This isn't the forum to describe them but if you are curious, email me.

- Line 3 - CHS stands for Cylinder, Head, Sectors and describes the HDs geometry. It also tells you the technical aspects of the geometries.

- Line 4 - Tells you the HDs caching system, the size of the cache, the HD's maximum number of sectors or BLOCKs per interrupt, and the current BLOCKs per intertupt setting.

- Line 5 - The HD is in 16bit mode, the current PIO or Programmed I/O data transfer mode is Mode1. I'm not sure what the (medium) means though. It also says that this drive DOES support DMA and the max supported *DMA* mode is Mode2.

- Line 6 - This tells you what Linux is using for the HD geometry (yes, it can be different than the actual HD's geometry). It also counts the total number of HD sectors, the HD is running in Logical Block Addressing (LBA) mode, and the total number of LBA blocks.

NOTE: LBA mode is critical for HDs bigger than 528MB to be properly used.

- Line 5 - This mentions the technical DMA timing requirements and the possible DMA modes.

- Line 6 - Mentions that this drive supports the legacy IORDY ISA line, the IORDY timing requirements, and finally, the maximum supported PIO mode.

Whew! Get all that? Hehehe.. don't worry about it. All you'll really care about is the 16/32bit mode, the PIO mode, and the DMA mode. Ok, so what settings is my drive currently using? Lets see:

```
                /sbin/hdparm -v /dev/hda

                        /dev/hda:
                         multcount    =  0 (off)
                         I/O support  =  0 (default 16-bit)
                         unmaskirq    =  0 (off)
                         using_dma    =  0 (off)
                         keepsettings =  0 (off)
                         nowerr       =  0 (off)
                         readonly     =  0 (off)
                         readahead    =  8 (on)
                         geometry     = 525/64/63, sectors = 2116800, start = 0
```

So, you can see that multicount is OFF, 32Bit mode is ON, etc.

* Before you make any changes, do a quick NON-DESTRUCTIVE (ie. this won't hurt any of your data, etc) benchmark of your HD by doing:

```
                /sbin/hdparm -t -T /dev/hda

                        /dev/hda:
                         Timing buffer-cache reads:   32 MB in  1.82 seconds =17.58 MB/sec
                         Timing buffered disk reads:  16 MB in  7.27 seconds = 2.20 MB/sec
```

As you can see, the top figure is really a benchmark of all of the system's memory, caching, etc. (This is slow since this is only a 486-160). The second benchmark is the actual HD's read performance. Again.. this is VERY slow since I only have a Mode1 IDE controller in this system. Doh! Also, if your system has 8MB or less memory, the benchmarking might not work for you.

- Ok, so lets TUNE this thing!

NOTE: As you start trying to use these HDPARM commands, your system might freeze (crash). If it does, you can pretty much count on that your system not being able to run with that option. I too have done this a few times and everything came back up after a RESET. Your milage will vary and you might lose data though I have been lucky. So, first, read item #1!!!

1. Be VERY sure you have a good tape, CD-R, etc backup of your machine first. Reading the hdparm man page warns that some of these settings on a not-so-well IDE subsystems destroy your HD's data. So, YOU'VE BEEN WARNED!

2. Ok, turn on 32bit transfers on your first HD by typing in "/sbin/hdparm -c3 /dev/hda". Once that is turned on, benchmark your HD again with the "/sbin/hdparm -t -T /dev/hda". Any improvement?

NOTE: On my 486 system with the lame IDE controller, I didn't see much.

3. Next, turn on the HD's blocking mode if it isn't currently already set. To do this, get the max blocking mode using "/sbin/hdparm -I /dev/hda" and then enter in the MaxMultSect number into the command (example here is 16):

```
                              "/sbin/hdparm -m16 /dev/hda".
```

After this setting, rebenchmark your HD.

NOTE: As mentioned in the hdparm man page, some drives will actually run SLOWER with higher BLOCK modes. Because of this, I recommend that you try multiple sizes and then re-benchmark the HD.

4. Lastly, for those of you with Mode3/4 IDE controllers with EIDE or UDMA HDs, enable the Mode3 or Mode4 PIO and UDMA mode if it isn't already set.

NOTE: This is where it begins to get risky. I truly recommend that you read the hdparm man page on the -d, -p, and -X options.

If you are ready to try it, run the command:

```
                                      "/sbin/hdparm -d1 -X34 /dev/hda"
```

Now re-run the system benchmark and see what performance differences you've gained. Once you have found the optimal performance and stability settings, you need to make sure that the settings are restored upon a HD reset and also a system reboot. To do this, I recommend to APPEND the following lines to your **/etc/rc.d/rc.local** file. Please note the top line is ONLY an example and will need to be replaced with your optimal settings:

```
#Enable the kernel to perform optimal IDE I/O
/sbin/hdparm -d1 -X34 /dev/hda

#Save the HDPARM settings over a HD reset
/sbin/hdparm -k1 /dev/hda
```

---

50 SPAM: Dealing with it and helping others stop it

This section has two pieces:

- SPAM
- Web Crawlers:

50.1 SPAM:

As you add WWW pages to the Internet, post messages to UseNET newsgroups, etc, you will find yourself getting MORE and more SPAM email. One or two SPAMs a week is ok (I suppose) but once you start getting 10+ a week, you'll get annoyed.

```
First, a few things should be understood about SPAM:

        1. When you receive a SPAM email, the SENDER almost never use
           their own email servers to send them out.  They are usually using
           someone else's mis-configured email MTA (mail transfer agent) to do
           it.  You might think this isn't that big of a deal but consider:

                A. it is filling up the innocent email relayer's internet connection
           with SPAM traffic that has NOTHING to do with their normal business.

                B. for each email the SPAMER sends to this relay site, thousands
                   to tens of thousands emails leave.  This saturates the email
                   server, its overall performance, etc.

                C. The innocent email relayer's entire Internet domain could be
           blocked from the internet via the various anti-SPAM systems
           (RBL, ORBS, etc) because they have been spamming people.
```

Ok, so say you got a piece of SPAM. How can you tell what is really going on? Here is one SPAM I received that I'll use as an example. Bare with the length here but its important to see ALL of their various tactics:

```
        1. If you were to simply REPLY to this "FROM" address, the email
           would bounce because it is forged (totally bogus).

        2. The only way to get a hold of these people is to call some toll
           free number.

        3. SPAMs sometime say this email meets "compliance with the proposed
       Federal legislation".   Why?  Because they offer a way to
```

```
       unsubscribe from from their list.  But..

                A. They usually use those free internet email services
                   out there (hotmail, yahoo, etc) to do this.  Not their real
                   email addresses so when those sites ARE put up, they are usually
           shut down quickly as all the free services out there strictly
           prohibit spammers from using their services.

                B. They never read the complaints the receive but they DO use those
                   hate emails to confirm that your email address is VALID.  Once they
                   know your email address is valid, they either send more spam to
                   you or sell your address to some other spammer.

                  ** This is why its CRITICAL to NOT to EVER email these addresses **

                C. By using these free email services, the spammers are breaking those
                   service's Anti-SPAM rules.
```

The email without full headers:

```
----------------------------------------------------------------------------
From: "Barbara23347@powerworx.net" <Barbara23347@powerworx.net>
Subject: Dental & Optical Plan Savings - Limited Time Only
Date: Wed, 21 Oct 1998 06:15:00 -0400 (EDT)

Hello,

We work with a group of your local doctors and dentists
and are offering a Dental - Optical Plan that runs
approximately $3 a week for an individual and
4 a week for the entire family with no limit to the number
of children.

Would you like our office to furnish you with the details?
Call Toll-Free

1-800-929-7648
"Refer to the K601 offer." (be sure to give this)


*If your state is listed below then we currently do not
service your area.

************************************************
We are linked to plenty of web sites that offer
free subscriptions to our mailing list.
You may JOIN or LEAVE this list at
any time by following the simple instructions that
can be found at the end of this email.

You are on our mailing list
```

```
because you have subscribed at one of our
associate web sites, sent us email or we have a previous
online relationship.

Marketing Service Co.
Customer Service Department
1-913-562-0134

This message is being sent to you in compliance with the proposed Federal
legislation for commercial e-mail (S.1618-SECTION 301).
"Pursuant to Section
301, Paragraph (a)(2)(C) of S. 1618, further
transmissions to you by the sender of this e-mail may be stopped
at no cost to you by clicking <A HREF="<url url="mailto:kppt@mypad.com">here</A>">; and
placing REMOVE in the subject.</FONT></CENTER>

*************************************************
------------------------------------------------------------------------------
```

Ok, so where did this email REALLY come from and how can you STOP this SPAM in the future?

```
        Well, first, you need to enable your email reader to show the FULL EMAIL
        HEADERS.

                Pine:

                        Go to the main Setup-->Config menu and enable the following
                        commands:
```

```
                                enable-aggregate-command-set
                                enable-full-header-cmd
                                include-header-in-reply
```

```
                        Now, when you read an email, hit the "H"eaderMode or
                        "h" key and you will see the FULL headers.

                Eudora:

                        Click on the "Blah..Blah..Blah" icon


Now, here is that SAME email with full headers shown below:

        1. Little different eh?  Confusing even.  Which site actually SENT this
           email?  Was it someisp.net, mailcity.com, popsite.net, or powerworx.net?
           First, the various lines like X-Persona and other X-stuff don't really
           matter.  They are there more for information reasons.  You really want
       to look at the "received" line.  Ok, for the following
           example, there are TWO Internet domains of concern.  Usually, you won't
```

see two domains like this but BOTH are valid. This particular email server is configured to send/receive for both mailcity.com and popsite.net.

The email with full headers:

```
----------------------------------------------------------------------------
X-Persona: <someisp.net>
Received: from mta-mail.mailcity.com (02-070.038.popsite.net
209.198.10.70])
by someisp.net (8.9.3/8.9.3) with SMTP id DAA16082; Thu, 9 Sep 1999 03:18:16
-0700 (PDT)
Message-ID: <Mr3y0.fZpgJrR.4mmQHYk3mWcOXRBx.@mta-mail.mailcity.com>
From: "Barbara23347@powerworx.net" <Barbara23347@powerworx.net>
Subject: Dental & Optical Plan Savings - Limited Time Only
Date: Wed, 21 Oct 1998 06:15:00 -0400 (EDT)
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset="US-ASCII"
Content-Transfer-Encoding: 7bit
X-UIDL: fcfe6e177a9ad2665891d53ba4e141aa

Hello,

We work with a group of your local doctors and dentists
and are offering a Dental - Optical Plan that runs
.
.
.
----------------------------------------------------------------------------
```

So, now what?

Well, you need to take this email with FULL headers and forward it to the correct people. For this example, I emailed:

abuse@popsite.net, postmaster@popsite.net, abuse@mypad.com and postmaster@mypad.com

1. Why use the "popsite.net" address over the "mailcity.com" address? No reason, either would have worked.

2. Why the abuse and postmaster addresses? The abuse address is well known for notifying remote sites about SPAM problems. The postmaster address is well known as the address for the email server administrator.

3. Why the mypad.com address too? I also email these these people because ANYONE associated with SPAMMERS will almost ALWAYS discontinue the spammer's account. This is a very effective way to shut spammers down.

```
                From here, I recommend to prepend the original spammer's
                subject field with "SPAM:" and also to start the email
                body off with something like:

                --
                Spam Alert:

                        popsite:        You are relaying spam.  Please fix your MTA
                        mypad:      Please delete this account


                Then add the original spam email with ALL the headers.
                .
                .
                .
                --

--

That's it!  You will probably get an automated email back from the various
sites letting you know you that they received your email and they will act
upon it.  Some sites will personally email you back telling you that they
dealt with it.


So, that's it.  Right?  NOPE.

        Many of these sites will still relay email for spammers though
        you've ASKed and asked them to stop.  What to do?

                Report them!  To who?


        Go to these recognized Anti-SPAM sites:
```

```
                        Is the relay already filtered:                  Report it:
                        ------------------------------                  ----------
        RBL:            <url url="http://maps.vix.com/cgi-bin/lookup">                   http://maps.vix.com/rbl/r
        Orbs:           <url url="http://www.orbs.org/verify1.cgi">                     http://www.orbs.org/email
        IMRSS:  <url url="http://www.imrss.org">                                        http://www.imrss.org/cgi-
        IMRSS DSSL:     <url url="http://www.imrss.org">                                        http://www.imrss.
        RRSS:           <url url="http://relays.radparker.com/nph-lookup.cgi">  http://relays.radparker.com/nph-s
```

P.S. Be SURE that you are using some of these filtering systems via your Sendmail setup. Check out the Sendmail section 25 (Section 25), for more details.

50.2 Web Crawlers:

If you get several firewall hits that looks like:

```
--
Sep 12 11:15:13 roadrunner kernel: IP fw-in rej eth0 UDP 209.249.159.162:137 100.200.0.0:137 L=78 S=0x00 I=32141
--
```

Try TELNETing to that site. You will then see:

```
--
[root@roadrunner]# telnet 209.249.159.162
Trying 209.249.159.162...
Connected to 209.249.159.162.
Escape character is '^]'.
UNAUTHORIZED ACCESS!!!
You are not authorized to connect to this host.
Violations will be prosecuted to the full extent of the law.

See <url url="http://www.scour.com/General/Misc/Add_Or_Remove_Site.phtml"> for information on removing your host

Connection closed by foreign host.
--
```

What the hell is this? It's a web crawler (Spider) that is trying to index everyone's insecure Microsoft File & Print shares. Personally, these people make me sick by doing this but they DO allow you a way to disable it. Go to the URL shown above and remove your box from their SMB crawler.

51 FS Recovery: How to fix LILO and file system problems

Lets say that one day, you have to reboot your machine to install new hardware, find your machine CRASHED, etc. Upon reboot, you see an error like:

- LI (LILO never fully loads.. it just sits there)

or

- The kernel loads up fine but then says: "Vfs cannot open root device 08:11 kernel panic :vfs:unable to mount root fs on 08:11"

–

First, ask yourself:

A. What has changed recently? Did you add/remove any hard drives recently? Keep this in mind:

With IDE drives, they ALWAYS get the same name. IDE0-Drive0 is always /dev/hda and IDE1-Drive1 is always /dev/hdd.

With SCSI drives, they get their name dynamically. So if you have drives on SCSI ID 0, 4 and 5, you would have /dev/sda, /dev/sdb, and /dev/sdc (NOTE the lack of correspondance from the SCSI ID # and the drive name). NOW, lets say ID #4 DIED. Upon reboot, you would NOW see /dev/sda and /dev/sdb. Notice that old /dev/sdc is now "b". Sucks huh? This really can screw things up, especially for software RAID setups!!! Hopefully, this naming issue might be fixed in the 2.4.x kernels.

B. What drive do you boot from?

/dev/hda or /dev/sda

C. What drive is your / partition on?

/dev/hdaX, /dev/sdaX, etc

** For this example, I'm going to assume /dev/hda5 **

First, create a set of Linux RESCUE diskettes. This is done using "RAWRITE" or "dd" from images on your CDROM, an FTP server on the Inet, etc. You will need the BOOT and RESCUE images put onto diskettes.

Next, after you load up the rescue disks:

1. Mount your suspected "/" [root] partition

(mkdir /mnt/mnt; mount -t ext2 /dev/hda5 /mnt/mnt) Is everything there in /mnt/mnt as you expect?

A. No? Make sure you mounted the right partition. If you are *SURE* this is the right partition, umount this partition (umount /mnt/mnt).

Run "fdisk /dev/hda" and make sure all your partitions are there. If they are good. If they aren't, umount this partition, reboot and go into the CMOS setup.

Now, make SURE that your CMOS setup for the HDs (number of cylinders, heads, sectors, TRANSLATION) is configured the SAME way as when you installed Linux. I have seen a few times where the TRANSLATION settings were toggled from LBA to NORMAL or AUTO was being unreliable. For large HDs (> 1GB), it should be set to LBA.

NOTE: I do NOT recommend the use of "AUTO".

Upon reboot, re-run fdisk and hopefully your partition tables are ok. If not, I hope you documented your partition tables much like I did in the first chapters here in TrinityOS. If you didn't, you have a few last options.

Email me and I can give you some notes on how to rebuild a FS from the SuperBlocks or you can try some of the tools below. Please note that these tools might not be around anymore or there are now newer/better ones. If you know of other disk tools for Linux, please let me know.

Thanks to Harondel Sibble for this list ———————————————————— (i) findsuper is a small utility that finds blocks with the ext2 superblock signature, and prints out location and some info. It is in the non-installed part of the e2progs distribution.

(ii) rescuept is a utility that recognizes ext2 superblocks, FAT partitions, swap partitions, and extended partition tables; it prints out information that can be used with fdisk or sfdisk to reconstruct the partition table. It is in the non-installed part of the util-linux distribution.

(iii) fixdisktable (`<http://bmrc.berkeley.edu/people/chaffee/fat32.html>`) is a LINUX utility that handles ext2, FAT, NTFS, ufs, BSD disklabels (but not yet old Linux swap partitions); it actually will rewrite the partition table,if you give it permission.

(iv) gpart (`<http://home.pages.de/michab/gpart/>`) is a utility that handles ext2, FAT, Linux swap, HPFS, NTFS, FreeBSD and Solaris/x86 disklabels, minix, reiser fs; it prints a proposed contents for the primary partition table, and is well-documented. Recommended! ————————————————————

Reboot into the rescue disk and try again. If things still aren't right, you are in a last ditch situation. The filesystem is probably a mess. Cross your fingers NOW and follow the next step.

B. Yes? Now unmount it (umount /mnt/mnt) and run a file system check on it. (e2fsck /dev/hda5) Make sure everything is cleaned up. You might be prompted if you want to fix things along the way. Say "Yes".

If "e2fsck" it cannot complete, email me again and I can tell you how to do some final last tricks before you have to just format and restore from tape or completely re-install the OS.

C. Remount the / partition as show in A.

2. In /mnt/mnt/etc/lilo.conf, make sure that the "boot" line points to the correct boot drive (boot=/dev/hda).

NOTE: there should not be any NUMBER after the drive letter. This means its using the Master Boot Record or MBR to boot.

3. In the TOP most "image" section, make sure that:

- the specified "image" file exists in /mnt/mnt/boot - the specified "root" line is your actual partition for the / drive. - Exit out of the editor and save any changes

4. In /mnt/mnt/etc/fstab, make sure that the line that has the "/" in the second column reflects the correct drive and partition of your / partition. You should also confirm this for the possible other partitions like /var, /usr, /tmp, /home, etc.

5. Ok, here comes the magic if you DID make any changes to /etc/lilo.conf, run the following command from the rescue diskette

lilo -C /mnt/mnt/etc/lilo.conf -r /mnt/mnt

If everything goes well, you should see LILO run and print out all of your configured kernels with the top-most one with a "*" next to it.

6. Reboot and hopefully things are ok now.

52 Gracefully transitioning Internet domains through a IP address or ISP change change

Changing IP addresses and/or ISPs soon?

Making a smooth transition from one IP address to another isn't too hard though you need to do some proper planning and configuration ahead of time.

Here is a check list you need to do IN order:

Before you move: ————–-

1. Arrange with other sys admins to be both a backup DNS and SMTP server for you (they don't have to be the same machine or even service provider). I recommend to have at least (2) backup DNS servers and (1) SMTP server that are connected via entirely different ISPs. Setting up both backup DNS and SMTP servers is covered in their respective TrinityOS sections.

2. Next, you need to update your Internic registrar (Network Solution is one example). You need to tell the Internic your new backup DNS servers. Do this quickly as it takes time and some registrars constantly screw things up OVER and OVER and OVER.

3. Configuring backup SMTP is a matter of setting up an extra higher cost MX record(s) in DNS and adding your domain name to the /etc/mail/relay-domains file. Make sure you test this backup email mechanism as well. This will be added to trinityOS in the future.

4. Once you have #1 and #2 done, you need to change the DNS TTL (time to live) field in all of your domain zone files.. In each of your DNS zone records in /var/named, you need to change the TTL cache expiration # (last number in the SOA record). TrinityOS uses a TTL of "1D" or 24hrs. Change

this "1D" to "60" (seconds) for ALL your domain name records and also change the serial # to reflect today's date. Restart named (/etc/rc.d/init.d/named restart) and wait 1 day until all the various DNS servers on the Internet time out your old cache settings. About to shut down your old IP address (24hrs after task #4): ————————————————————-

5. Go to your Internic registrar and update your account to reflect your new TCP/IP address for your main server. For Network Solutions, you should use their "host" form. Do NOT proceed until you get a notice back from your registrar that they have accepted your changes. Also note that though they might update your records, a "whois" might not reflect the changes as quickly as a "nslookup".
6. Once you have confirmed that the Internic has your new TCP/IP address, edit your various domain zone files in /var/named and change both the serial # to today's date AND change the TCP/IP address of your main NS record to reflect your new IP address.
7. Copy the old reverse DNS zone file for your old reverse IP zone file and now create a new reverse IP address zone file to reflect your new IP address.
8. Next, update the /etc/named.conf file to reflect the new reverse zone's filename from step 7.
9. Restart named (/etc/rc.d/init.d/named) to propogate your new zone files (w/ your new IP) to all the backup DNS servers). Changing your IP: ——————
10. Update /etc/hosts, /etc/hosts.allow, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth* (* = your external NIC), and /etc/rc.d/rc.firewall with your new IP address. Shut down your box ———————
11. Bring your box back up on the new network w/ the new IP
12. Have someone send you test email to make sure that DNS and email is working ok.
13. Finally, if everything is ok, re-edit all your domain zone files and update both the serial # and change the TTL back to 1D. Don't forget to restart named so both your DNS server and all your backups are updated.
14. Finally, make sure that all of your backup DNS servers accept new zone file xfrs from your new IP address. This security measure is controlled by their /etc/named.conf file.

53 Setting up Linux as a good desktop operating system

As Linux distributions get better and better, Linux is truely becoming a very usable desktop operating system. But, there are some tools out there that are still better than their OpenSource alternatives. Specifically, I recommend that you install the following tools on your system.

- OpenSSH or SSH.com's SSHv2 for secured telnet and file transfers
- Sun's StarOffice or OpenOffice.org's suite for word processing, spreadsheets, etc.
- Sun or IBM's Java run-time or full JDK for Java support
- Mozzila for a excellent WWW browser and a strong Email client
- Adobe Acrobat to view PDFs as they really should look
- XMMS for MP3 and other audio file playback
- RealAudio's RealPlayer for both audio and video

- Setiathome - use spare CPU cycles to search for E.T.
- GAIM - One of the best ICQ/AIM/Yahoo/etc Instant Messanger (IM) clients available
- xine - To watch DVDs under Linux

If you have a "minimum required" program for your Linux box, please email me and let me know what it is. If enough people request it, I can touch on the installation of these programs.

54 Thoughts about the needs and procedures to Patching your Linux distribution

All users should apply patches to their respective Linux installation:

1. upon the first time the machine is installed

2. at least every week after that to stay ontop of the newest bug and security fixes

To find out what are the current security issues with Linux, etc, check out the Security URLs in 5 (Section 5)

— — —

NOTE: This is where Redhat RPMs, and Debian upgrade files really shine and blow away Slackware .PKG files!

NOTE #2: Be careful of where you download your newer versions of source code, RPMs, etc. Recently, `<ftp://win.tue.nl>` was hacked and the hackers put trojan'ed versions of TCP-wrappers and Linux-utils on their site. Because of this, many user's passwords were sent to the hacker's email address, etc. Not good.

In the future, I will cover how to verify the package's authenticity with PGP.

Redhat users: Depending on when you purchased your CD, your CD might already have these RPMs installed so if it says the RPM is already installed, just skip it.

************************* ** Be cautious with RPMs ** Before you blindly start installing new patch RPMs or even new software in RPM form, you really should (quickly) inspect the RPM archive to make sure it looks ok. For example, lets say you are going to install a new Sendmail RPM:

First, download the new Sendmail RPM file and put it to some location for future reference. I personally put all files in /usr/src/archive as described in the top of 5 (Section 5)

Now show the RPM creator's notes:

```
rpm -qip sendmail-*.i386.rpm
```

Show the RPM's file contents:

```
rpm -qlp sendmail-*.i386.rpm | more
```

- Next, if you already have an older Sendmail RPM installed, make sure that the new RPMs won't clober your old configuration files:

```
rpm -Uv --test sendmail-*.i386.rpm
```

For even more info (I'd recommend it), do:

```
rpm -Uvv --test sendmail-*.i386.rpm
```

- With a little cautious looking, you'll know what will happen if you install this new RPM. Ok?

If the new Sendmail installation is going to copy over your original files, the RPM will -usually- make a backup of your configuration files and add a ".rpmsave" to it.

*** ******************************

Redhat users #2: I have noticed that the "rpm" program will crash (coredump) about 60% of the way through a wildcard (*.rpm) RPM upgrade process. You should be able to safely figure out what patches it failed to install and do them manually or by doing the following:

Say that the RPM program died while doing patching in the letter range (Q). So, do this to install all patches from Q to Z.

```
"rpm -Uvh [q-zQ-Z].rpm
```

*********************** ** Patching your Redhat system ** Now, to find out if any new RPM files exist for Redhat, go to `<http://www.redhat.com/support/docs/errata.html>` and then look at the upper right-hand corner's date. If this date is NEWER than the 00readme.errata file, then there are newer RPMs.

Their documentation system read SUCKS in terms of though there might be a NEWER RPM for Glibc, they mearly update the DATE in the previous Gblic errata entry. Lame eh? So, you will have to page though the different errata listing to find what newer-date entries have been added.

*** ******************************

55 Serial Linux Consoles and Reverse TELNET

One great thing about good rackmount PCs is their ability to be completely controlled via serial port and NOT require a VGA output and keyboard control. This is all done via the machine's BIOS and it just works. Unfortunately, if you're like me, you don't have a machine that supports this in the BIOS. Don't fret!

Linux also has the ability to display and manipulate the full LILO boot section process, show the full kernel bootup sequence, and ultimately allow for system login via any serial port. In addition to this, you can take any serial port and make it available as a Reverse TELNET port. Reverse TELNET is the same thing as console ports on terminal server such as a Cisco 2511, etc. You just telnet to a specific TCP port or a specific IP address on the Linux machine and you are then directly communicating to that other host via a serial port through TELNET. Very simple and a LOT cheaper than real terminal servers.

55.1 Lilo and Daemon Boot Logs via a Serial Port

Enabling LILO and boot logs via a local serial port is pretty simple. Modern Linux distributions should have this automcatically enabled but just in-case, follow these kernel compile-time options. After you have enabled these options, follow the instruction in 14 (Section 14: Kernel Compiling section).

The following example is for a Linux 2.2.x based kernel:

```
Character Devices

    --> Standard/generic (dumb) serial support
    ----> Support for console on serial port
```

Optionally, if you are trying to use a Multi-port serial card like a Cyclades unit, simply enable it under a the same kernel configuration section:

```
Character Devices

    --> Non-standard serial port support
```

If you are trying to setup a Reverse TELNET server, you'll need one of these higher density serial cards if you want to control more than one or two serial devices. I'm using a Cyclades card without any major issues.

Anyway... once you configured/compiled/booted your new kernel (if required), you then need to edit the lilo.conf file.

NOTE: This config assumes the use of COM1 running at 9600Kbps, No partiy, 8 bit / 1 start bit / 1 stop bit. Other serial ports like Cyclades ttyC* are legal as well as other serial speeds and settings.

/etc/lilo.conf

```
#This puts LILO over the serial port - this is an interactive prompt if desired
serial=0,9600n8


#The following sends the kernel boot messages to --BOTH-- the serial port
#  and the console CRT screen.  The system daemon bringup logging is
#  --ONLY-- sent to the console CRT screen.
#
# I recommend this setting
#
append="console=ttyS0,9600 console=tty0"


#Like above, the kernel messages go to the console CRT and serial port.  But
#  now, the system daemons bringup logs now --ONLY-- display to the serial
#  port.
#
#   -- If you are aware how to send the system daemon bringup logs to both
#      the CRT and serial port, please email me.
#
# Disabled by default.
#
#append="console=tty0 console=ttyS0,9600"
```

That's it. Just re-run "lilo" as root and make sure LILO run cleanly.

Ok, one more step. You need to enable the "login" daemon on this serial port. To do this, edit the `/etc/inittab` file and find the lines that look like:

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
```

```
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

At the end of this text section, add the line:

```
7:2345:respawn:/sbin/mingetty ttyS0
```

Please note the unique number "7" at the beginning and the updated TTY port "ttyS0". Save that file and restart the "init" process by running the command `telinit q`.

That's it! Just to make sure things are running correctly, run a serial COMM program on another machine that will be communicating over the serial connection. Make sure the port, speed, etc. is all correct and just hit the ENTER key a few times? Hopefully you will see a login prompt from the Linux host.

As a final test of everything, reboot the Linux machine and watch the LILO, possible kernel logs, and login prompt show up at 9600 baud.

As like all TrinityOS sections, I don't go very deep into troubleshooting things. If you need more detailed help, please see 5 (Section 5 - Serial Consoles and Reverse TELNET) for additional help URLs. If you are still stuck, feel free to send me an email.

55.2 Reverse TELNET terminal services

Terminal servers are great for controlling remote network devices, etc. All you do is TELNET to some IP address or some specific TCP port on a particular IP address and you are then transparently communicating to a different device via it's console (serial) port. Unfortunately, terminal servers like Cisco 2500s, Livingston Portmasters, Cyclades, etc. are expensive. Fortunately with the use of a multi-port serial card from vendord like Cyclades, Digi, etc., you can turn a Linux server into a Reverse TELNET device very cheaply:

For this documentation, it assumes the following (PLEASE READ):

- The GNU UUCP package needs to be installed. Technically, all we need is the "cu" program. There might be better alternatives to "cu" and I'm very open to alternatives.
- You TELNET to a specific host IP address using different TCP ports for different console ports
- Connectivity is available via ALL network interfaces on the Linux server (0.0.0.0:*). You can disable any number of interfaces from listening to Reverse TELNET requests if you wish (say to only allow connections from 127.0.0.1)
-
- IP 192.168.0.1 - TCP port 300 - connects to /dev/ttyC0
- IP 192.168.0.1 - TCP port 301 - connects to /dev/ttyC1
- ..
- IP 192.168.0.1 - TCP port 307 - connects to /dev/ttyC7
-

Currently, the knowns issues with this method are:

- There is currently NO authentication or encryption when connecting to the remote serial console port

- cu's escape mechanism doesn't work. It should also be noted that cu's default escape sequence is the same as the one use in SSH.
- It seems that some serial output is conflicting with cu and thus cu is not showing this output. I'm looking into this issue now (01/10/03).

Ok, getting down to it. Make sure that the serial card is installed and working in desired the Linux machine (not currently covered in TrinityOS). I recommend to use a COMM program like Linux's "minicom" to verify that the card and serial cabling is working correctly FIRST.

For example, the Cyclades Cyclom-Y 8-port serial card uses ports /dev/ttyC0-7. To test, load up Minicom, change it to use the proper serial port ("Control-A", "o" - for Options, "Serial Port Setup", and change the port, speed, etc.). Once changed, save your settings as "dfl" (default), exit out of Minicom, and reload it. Hit enter a few times to make sure you get a login prompt.

So, first thing to do is register these new TCP ports. Please note that I've used ports TCP ports 300 through 307. These are legal available ports accordind to the IANA but you can use anything you'd like. Just make sure something else isn't using your proposed ports first (run "netstat -an" to check).

/etc/services

```
# Local services
console0        300/tcp       # Reverse TELNET console service - TrinityOS
console1        301/tcp
console2        302/tcp
console3        303/tcp
console4        304/tcp
console5        305/tcp
console6        306/tcp
console7        307/tcp
```

Next, we will use XINETD to start and re-start the individual ports when under use. If you would like to see INETD examples, let me know via email.

NOTE: You will need to re-create each individual files /etc/xinetd.d/console0 through console7 from this one example. Please also be sure to change the "console0" and "ttyC0" text to reflect the proper XINET service and serial port.

/etc/xinetd.d/console0

```
# default: off
# description: The reverse telnet console server serves console sessions via
#              telnet sessions; it uses unencrypted communications and is NOT
#              authenticated.
service console0
{
         flags          = REUSE
         socket_type    = stream
         wait           = no
         user           = root
         server         = /usr/bin/cu
         server_args    = -E+ -l /dev/ttyC0 -s 9600
         disable        = no
}
```

Ok, once you created all 8 files (in this example for a Cyclades 8-port card), restart XINET by running:

```
/etc/rc.d/init.d/xinetd restart
```

So, that should be it. From the server, try it out:

- telnet 192.168.0.1 300
- telnet 192.168.0.1 301
- etc.

To disconnect, just use TELNET's <Control-]> escape sequence, then type in "close" and that's it!

56 Common Observations, Q&A, etc

```
#1      - SYSLOG:       Many users notice that they get "--MARK--" messages in their SYSLOG files.
                        Why?

                A:      This is a feature of SYSLOG to let you know that its still working, though
                        it has nothing to report.  If you don't like this behavior (or it was
                        automatically enabled via a RPM update, etc), edit its loading to be something
                        like "syslogd -m 0"

                                Redhat: edit the /etc/rc.d/init.d/syslog file

                                Slackware:      edit the /etc/rc.d/rc1.inet file


#2      - SYSLOG:       Many users notice that they sometimes get the following message:

                        "May  2 04:02:21 rocko kernel: klogd 1.3-3, log source = /proc/kmsg started.
                        May  2 04:02:21 rocko kernel: Inspecting /boot/System.map
                        May  2 04:02:22 rocko kernel: Loaded 4253 symbols from /boot/System.map.
                        May  2 04:02:22 rocko kernel: Symbols match kernel version 2.0.36.
                        May  2 04:02:22 rocko kernel: No module symbols loaded."

                        What is this from?

                A:      This is from Redhat's "logrotate" program restarting the SYSLOG service.  No
                        worries.. this is normal.
```

57 ChangeLOG

```
+------------------------------------------------+
|  Notice to all TrinityOS viewers:              |
|                                                |
|   - If there are any sections that you would   |
```

```
|     like to be added/modified/corrected, etc,    |
|     just let me know!                            |
|                                                  |
|  ** Do you want to get an e-mail when I          |
|     update the TrinityOS doc?  Just send an      |
|     e-mail to dranch at trinnet dot net with a   |
|     subject of "Add me to your updates list" and |
|     I'll add you to the list!  **                |
|                                                  |
|                      dranch at trinnet dot net  |
+--------------------------------------------------+
```

See all prior updates older than 01/12/03 at:

`<http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS/UPDATES/TrinityOS-old-updates.txt>`

```
**************************************************
**  TrinityOS                                   **
**              "CRITICALITY" list              **
**************************************************

- This section is for TrinityOS users to better track what TrinityOS
  changes ARE and AREN'T so IMPORTANT to be fixed on their Linux box


Key:
----

*C =    CRITICAL:

                Something CRITICAL means that your are vulnerable to
                attack either due to some new security exploit, an
                error on my part (firewall rules, etc), or something
                that should be tested ASAP.


I =     IMPORTANT:

                Something IMPORTANT means that these changes will
                have direct impact on the functionality of your box
                or is a medium security risk.  Not all IMPORTANT things
                are important to everyone.


G =     GOOD READ:

                Something as GOOD READ means that it is informative
                and will better help you track your machine.


N =     Not Important:
```

```
                        Something NOT IMPORTANT are things like Typo corrections,
                        formatting changes, etc.

===============================================================================
Criticality
--
          Date          What was changed and in what [Section]
      --------  ------------------------------------------------
===============================================================================



-----------------------------------------------------------------------------
All of TrinityOS's step-by-step instructions, files, and scripts are fully
scripted out for an automatic   installation at:

http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-security/TrinityOS-security.tar.gz
----------------------------------------------------------------------------

N   05/22/05    - Updated various programs to their newest versions
    * Sent        [Section 5 - URLs]
      Update *
                - Cleaned up the ssh section a little
                  [Section 30 - SSH]

-----------------

G   04/16/05    - Updated the IPCHAINS firewall to 4.21 where I updated the
                  bogon list to reflect changed bogon listing and added
                  output Multicast and NFS traffic filters

-----------------


N   02/25/05    - There was a typo in the IANA assignments URL for the wget
                  line compared to the raw URL.
                [Section 5 - URLs]

-----------------

G   07/31/04    - Fixed the lock entry to point to /var/lock vs. /var/log
                  Thanks to Bill Marr for this one.
                [Section 36 - UPSes]

-----------------

N   07/26/04    - Updated the example host name for finding out the Bind
                  version from @xyz.com to @ns1.xyz.com.
                [Section 24 - DNS]
```

```
-----------------
N   07/24/04    - Updated the kernel versions:
                  2.6.x  --> added 2.6.7
                  2.4.22 --> 2.4.26
                  2.2.25 --> 2.2.26
                  2.0.39 --> 2.0.40

                - Updated the apcupsd website url and version
                [Section 5 - URLs]

-----------------

N       07/13/04    - Updated the ISC DHCPd server version to 3.0.1rc14
                [Section 5 - URLs]

G               - Updated the Linux distribution section a bitA
                  - Added a RPM list that is offered in RHEL ES 3.0
                [Section 6 - Distros]

G               - Updated the DHCPd configuration to reflect 3.0.1rc14
                    - Updated 255.255.255.255 route requirement is for 2.0.x
                      and 2.2.x kernels
                    - changed location of the dhcpd.lease file from /etc to
                      /var/dhcpd/
                [Section 27 - DHCPd]

-----------------

G   03/21/04    - Updated the sendlogs section to 03/14/03 which includes
    * Sent        log reduction.  Specifically, many users get lots and
      Update *    LOTs of firewall hits but they might not care about say
                  port 80.  Sendlogs now counts the # of hits and deletes
                  them out of the email so you can more quickly scan your
                  logs email.  I've been using this for a long time now and
                  it's a VERY nice feature.
                [Section 9 - Adv. System Logging]

-----------------

G   03/14/04    - Added the backup-to-disk script to support both local and
                  remote NFS / SAMBA backups to hard drives.  This includes
                  both internal as well as firewire and USB HDs.
                [Section 29 - Backups]

-----------------

G   02/29/04    - Added a wget command to download a local IANA list
                [Section 5 - URLs]

-----------------
```

```
G   11/21/03    - Clarified that cutting and pasting TrinityOS scripts from
                  a web browser into a text file will most likely create many
                  errors.  It's ALWAYS recommended to get a copy of the
                  TrinityOS scripts via the TrinityOS-archive file.
                [Section 10 - Firewalls]

-----------------

N   11/10/03    - Updated / deleted all URLs that pointed to kernelnotes.org
                  Thanks to Jamie Alessio for the notice

-----------------

G   11/08/03    - Updated various daemon versions
    * Sent        - 2.4.22 is stable
      Update *    - bind 9.2.3
                  - bind 8.4.1
                  - sendmail 8.12.10
                  - dhcp 3.0p2
                  - wuftp 2.6.2 with many patches
                  - mozilla 1.5
                  - openssh 3.7.1p2
                  - raidtools 1.00.3
                  - samba 3.0.0
                  - apcupsd 3.10.6
                  - apache 2.0.48 and 1.3.29
                  - nmap v3.48
                  - gaim 0.72
                [Section 5 - URL]

                - Updated the versions of distros
                  - Mandrake 9.2
                  - SuSe 9.0
                  - Slackware 9.1

                - Mentioned that SuSe is being bought by Novell / IBM
                [Section 6 - distros]

-----------------

G   11/05/03    - Updated the distro discussion section about Redhat's
                  withdrawl from the basic enduser distribution business.
                  It also talks about their new Fedora project as well as
                  the various Enterprise Linux versions.  If you have questions
                  about RH EL, I have it running and can give you my thoughts.
                [Section 6 - Distros]

-----------------
```

```
G   10/05/03    - Updated the powerchute-generate-ups-graph.sh and
                  apcupsd-generate-ups-graph.sh scripts to fix an ellusive
                  decimal to octat conversion issue found in Bash.
                  Specifically, the script would throw errors like:
                  --
                  Filtering original powerchute.dat file..
                  Deleteing old ps and pdf files..
                  Creating files..
                      "generate-apc-graph-11003.gnuplot",
                          line 6: illegal day of month

                   - done creating files
                  Creating /tmp/ups-log-11003.ps..
                  Error: /undefinedfilename in (/tmp/ups-log-11003.ps)
                  Operand stack:
                  --
                [Section 31 - UPS]

-----------------

*C* 08/30/03    - Updated the Sendmail section to reflect that
    * Sent        relays.osirusoft.com is defunct and thus greatly slowing
      Update *    SMTP performance due to stalled DNS lookups for their
                  domain.

                  NOTE:  The loss of SPEWS isn't all that bad as they commonly
                         would block entire ISPs for a single spammer.  Not
                         very nice.

                  NOTE2:  Simply putting a "#" in front of the line:

                         FEATURE(dnsbl, `relays.osirusoft.com', \
                           `Rejected - See http://relays.osirusoft.com/')dnl

                         does NOT disable the use of osirusoft.  You must
                         DELETE the line, re-run the "generate-cf" script,
                         and then restart Sendmail for the changes to take
                         effect.

                [Section 25 - Sendmail]

-----------------

N   07/09/03    - Updated the SSH section to reflect OpenSSH and SSH.com
    * Sent        code versions 3.6.1p2 and 3.2.0
      Update *  [Section 5 - URLs]


G               - Updated the kernel compiling script "build-it" to
                  abort if the kernel image doesn't complete properly,
```

```
                  added the use of PATH variables, and added additional
                  ECHO statements for better compile tracking.  Changes
                  are also in the TrinityOS-security archive as well
                - I also updated the section's text to flow better, added
                  additional troubleshooting steps, etc.

                [Section 14 - Kernel Compiling]


G               - I wrote this up AGES ago but never added it to TrinityOS.
                  Anyway, I /finally/ added the installation of OpenSSH to
                  TrintiyOS and no longer recommend the use of SSH.com code
                  due to licensing prices.

                - Fixed a ssh typo where I was restarting syslogd and
                  not sshd (cut and paste error)

                [Section 30 - SSH]


N               - Renamed the TrinityOS-old-updates WRI file to TXT

N               - Moved all ChangeLOG entries older than 01/12/03 to
                  the TrinityOS-old-updates.txt file

                [Section 57 - ChangeLOG]

-----------------

G   06/24/03    - Fixed a typeo of /car/spool vs. /var/spool

                - deleted the incorrect restarting of the syslogd daemon when
                  it should have been crond.  Ultimately, this step wasn't
                  needed as cron will detect crontab changes automatically.
                  Thanks to LiNuCe for the report!

                [Section 41 - EXT2 tuning]

-----------------

N   06/12/03    - updated the IANA URL

                [Section 5 - URLs]

-----------------

N   06/07/03    - Updated the system info to reflect I'm running Mandrake 9.1
                  on the laptop (if anyone has questions about 9.1)

N               - Updated the Redhat versions from 7.1 to 9.0; Mandrake 8.1
```

```
                  to 9.1; Slackware 8.0 to 9.0; Debian 2.2R5 to 3.0R1;
                  SuSe 7.3 to 8.1; Added Gentoo

N               - Mentioned that the Corel and Storm distros are defunct;

N               - Mentioned which distros are community effort distros vs.
                  commercial ones.  Also mentioned that Caldera is now owned
                  by SCO; also added a note about their recent legal persuits

G               - updated my thoughts on RPM hell (it's not that bad now)

I               - Updated my thoughts on patch and errata support.  Specifically,
                  this was about my research on the Enterprise versions of
                  Redhat Enterprise and Mandrake Corporate server.

N               - Updated my thoughts on Mandrake's "drak family" utilities.

                - Some edits and distro update prods via Julian Buckley

                [ Section 6 - Distros ]

-----------------

N   05/17/03    - Added the recommendation to download ISC's PGP key
                [Section 5- URLs]

G               - Added PGP verification for Bind 9 source code
                [Section 24 - DNS]

-----------------

G   05/08/03    - The manual test of starting named still had the old Bind8
                  command line that included the old and wrong
                  "-g chroot-dns-int" syntax.
G               - Incorrect Redhat "chkconfig" command to make named start
                  after every reboot.  I was referencing "bind" instead of
                  "named".  It's now "chkconfig --level=345 named on"
                - Thanks to Nelson Rodriguez for top the bug report
                [Section 24 - DNS]

-----------------

N   04/08/03    - Update the kernel version to 2.2.25
    * Sent      - deleted the ICQ MASQ module sub-section as it isn't relevant
      Update *    for modern versions of ICQ
                - Updated samba to 2.2.8a to reflect new security issues
                [Section 5 - URLs]

G                               - Change the name of the section to now be "System Backups:
                  Recommended minimal file to floppy and using BRU"
```

```
                - Added the command to format the floppy
                - Change the MBR backup from going directly to the floppy to
                  /etc/info/mbr.dd
                - Added additional files tothe backup to the floppy:
                  fstab, raidtab, smb.conf(optional), smbusers (optional),
                  ssh2/ssh*, lilo.conf, resolv.conf, conf.modules, hosts,
                  hosts.*, inittab, dhcpd.conf (optional),
                  mail/*(optional)
                [Section 29 - Backups]

G               - Change the title to reflect only SSHv2 and not v1/v2
                - mentioned that tools are available to actively decrypt
                  SSHv1 traffic thus making SSHv1 basically useless
                [Section 30 - SSHv2]

*C*             - Updated the section to reflect that 2.2.8a is the current
                  secure version.
                - Updated the PGP key section to reflect that samba now signs
                  the tar files and not the .tar.gz or tar.bz2 files
                [Section 33 - Samba]

-----------------

*C* 03/29/03    - Yet another problem with Sendmail.  Updated the recommended
                  version to 8.11.7 or 8.12.9.
                [Section 5 - URLs]

*C*             - Updated the minimum version of Sendmail to avoid new security
                  issues.  HOW can Sendmail 8.12.x be chrooted but still have
                  two massive security expliots within weeks.  The new security
                  mechanism in 8.12.x is obviously flawed at best.
                - In the future, TrinityOS will move over to Postfix
                [Section 25 - Sendmail]

-----------------

*C* 03/28/03    - Updated the version of Samba to 2.2.8 to reflect a newly
                  fixed buffer overflow problem.
                [Section 5 - URLs]

*G*             - Updated the Samba section to reflect 2.2.8 and I also
                  improved the chapterization of this section
                - Added a specific code hack to help some users (utimes)
                  compile Samba
                [Section 33 - Samba]

-----------------

*C* 03/08/03    - Updated the version numbers of Sendmail to 8.12.8 and
                  8.11.6+ to reflect the recent remote root exploit issue.
```

```
                [Section 5 - URLs]

N               - Updated the version of Bind to 9.2.2
                [Section 5 - URLs]

G               - Updated the intro to reflect that Bind 9.2.2 requires a
                  non-vulnerable version of OpenSSL to be installed to support
                  DNSSEC.  TrinityOS doesn't cover this topic yet so this
                  issue is only mentioned.
                [Section 24 - DNS]

*C*             - Updated the versions numbers of Sendmail to 8.12.8 and
                  8.11.6+ to reflect the recent remote root exploit issue.
G               - Added an additional compiling recommendation to HIDE the
                  version of Sendmail you are running from the Internet.
                [Section 25 - Sendmail]

-----------------

G   02/22/03    - Updated the Copyright section to reflect some refined
                  wording, note TrinityOS's trademark numbers, and fixed
                  the URL pointing to the ultra-OLD .wri file.
                  Thanks to Simon Soltek for brining this to my attention.
                [Section 1 - Copyright ]

-----------------

I   02/18/03    - Updated the APCUPSd daemon to reflect 3.8.6 which fixes
                  a security issue
                [Section 5 - URLs]

-----------------

N   02/08/03    - Fixed some typos

                - Added XMMS and OpenSSH to the minimum recommended software
                  packages to install.
                [Section 53 - Minimum Recommended Software]

-----------------

N   01/31/03    - Updated the 3NIC IPCHAINS ruleset to add a missing
    * Sent        INT2BROAD varibale.  No worries, the correct settings are
      Update *    automatically used anyway.
                [Section 10 - Firewalls]

-----------------

G   01/26/03    - Added a URL for the Remote Serial Console HOWTO
                [Section 5 - URLs]
```

```
N               - Updated the Serial Console and Reverse TELNET section
                  to mention URLs in section 5.
                [Section 55 - Serial Consoles]

-----------------

N   01/13/02    - Updated the IPCHAINS rc.firewall ruleset to 4.10
                  - The latter half of the OUTPUT section was using
                    $UNIVERSE/0 instead of $UNIVERSE which was already
                    set to 0.0.0.0/0. This was a harmless typo and
                    didn't hurt anything but was incorrect.  Thanks to
                    Matteo Lunardi for catching this.
                [Section 10 - Firewalls]



********************************************************************************
* All prior updates dated 01/12/03 or older can be found  at:                  *
*                                                                              *
* http://www.ecst.csuchico.edu/dranch/LINUX/TrinityOS-old-updates.txt  *
********************************************************************************
```