# *Linux Journal* Issue #93/January 2002

LINUX JOURNAL



# Setting up a VPN Gateway

**Duncan Napier**

Issue #93, January 2002

How to install and run an IPSec-based VPN gateway with a firewall using a single bootable Linux diskettedistribution.

A virtual private network (VPN) is a tool that enables the secure transmission of data over untrusted networks such as the Internet. VPNs commonly are used to connect local area networks (LANs) into wide area networks (WANs) using the Internet. Perhaps you need to build a VPN between two offices but are not sure if the large infrastructure costs associated with an enterprise-level VPN solution are justifiable. The performance of applications that are intended for use over LANs (for example those that use network file sharing) seriously can be degraded over WAN connections. Likewise, lower bandwidth and longer latency in WAN connections can affect adversely the reliability and performance of groupware and thin-client applications. Perhaps you have a home office and would like to use your high-speed internet access to connect seamlessly and securely to your office LAN through an IPSec-capable router. Or perhaps you are just curious about VPNs and IPSec in general and want to experiment.

The VPN firewall discussed in this article will run on just about any 486-or-better PC that has 16MB or more main memory and two Linux-compatible Ethernet network cards. The idea is to provide a starting point from a single, self-contained package that will allow you to create robust, secure, scalable and highly configurable VPNs that also are interoperable with many common commercial VPN implementations. If you wish to experiment on a low-maintenance firewall-VPN gateway, then the package discussed here might be ideal for you.

This article shows you how to set up, at minimal expense, a working VPN gateway that uses the IETF's (Internet Engineering Task Force) IPSec (internet protocol security) specification. IPSec is an open standard and is supported by virtually all major firewall software and hardware vendors, such as Lucent, Cisco, Nortel and Check Point. This package will give you a widely interoperable

IPSec that uses the de facto standard 3DES encrypted, MD5-authenticated site-to-site or point-to-site VPN. You should be able to do this without resorting to a full Linux distribution or recompiling a standard Linux kernel with a kernel IPSec module.

The VPN system we examine here is based on FreeS/WAN (www.freeswan.org), a portable, open-source implementation of the IPSec specification. FreeS/WAN has been demonstrated to interoperate, to various degrees, with Cisco IOS 12.0 and later routers, Nortel Contivity Switches, OpenBSD, Raptor Firewall, Check Point FW-1, SSH Sentinel VPN 1.1, F-Secure VPN, Xedia Access Point, PGP 6.5/ PGPnet and later, IRE SafeNet/SoftPK, Freegate 1.3, Borderware 6.0, TimeStep PERMIT/Gate 2520, Intel Shiva LanRover, Sun Solaris and Windows 2000. The official FreeS/WAN web site has a regularly updated compatibility list with the latest version of its on-line documentation. FreeS/WAN version 1.5 is included in this package.

I have created a single-diskette distribution that installs the base configuration of a VPN firewall based on the Linux Router Project (LRP, www.linuxrouter.org), a compact Linux distribution that can fit on a single, bootable floppy diskette. The distribution here is essentially Charles Steinkuehler's Eiger disk image with Steinkuehler's IPSec-enabled kernel and LRP IPSec package. Firewalling is carried out through Linux ipchains. This particular version is based on the 2.2.16 kernel of Linux. This distribution is called DUCLING (Diskette-based Ultra Compact Linux IPSec Network Gateway). Compact Linux distributions have a twisted history. LRP technically refers to Dave Cinege's compact distribution. There are many variants around, including Charles Steinkuehler's distribution (EigerStein) of Matthew Grant's defunct Eiger version (lrp1.steinkuehler.net). Another such distribution is David Douthitt's Oxygen (leaf.sourceforge.net/ content.php?menu=900&page_id=1). Also, there is LEAF (Linux Embedded Appliance Firewall), a developer's umbrella that tries to coordinate releases and documentation, sort of like a one-stop shop for compact Linux distributions (leaf.sourceforge.net). I use the term LRP to refer to the compact Linux distribution presented here, even though some may consider this terminology incorrect.

If you are running MS Windows 9x, the distribution self-extracts and installs itself onto a standard 3.5", high-density floppy diskette. You also can write the image to a boot floppy if you have a system running Linux. Once the extraction is done, you will need to boot off the floppy disk you have created, copy the network drivers for your network cards over and edit the appropriate configuration files. That's it—no creating and formatting disk partitions or messing with boot managers on your hard drive. If you are not happy with the distribution, just pop the diskette out, throw it away (or reformat it) and reboot

your PC. Check the links on leaf.sourceforge.net/devel/thc for more information on these options.

### Background on the Firewall and the VPN

This distribution of LRP uses a standard ipchains-based firewall. **ipchains** (replaced by iptables in the 2.4 series kernels—see David A. Bandel's "Taming the Wild Netfilter", *LJ*, September 2001) is a freely distributed packet filter for Linux. It is very instructive to look through the ipchains HOWTO if you are not familiar with this firewalling tool. This can be found at www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html.

The VPN is provided by FreeS/WAN's implementation of IPSec. FreeS/WAN's IPSec implementation is compliant with the IETF's IPSec specification. IPSec is an extension to the Internet Protocol (IP) that provides for authentication and encryption. Three protocols are used to handle encryption and authentication, namely ESP (Encapsulating Security Payload), AH (Authentication Header) and IKE (the Internet Key Exchange). All these components are included in the FreeS/WAN implementation of IPSec and generally are transparent to end users. ESP and AH handle encryption and authentication, while IKE negotiates the connection parameters, including the initialization, handling and renewal of encryption keys. The only encryption scheme currently supported by FreeS/WAN is 3DES (the triple DES or Data Encryption Standard—the current de facto standard for IPSec encryption). Authentication is carried out using MD5 digests of a so-called shared secret (a shared key). The shared key could be a mutually agreed-to character string, RSA cryptographic key pairs or X.509 certificates. FreeS/WAN's KLIPS (kernel IPSec) component, which is compiled into the Linux kernel, implements AH, ESP and the handling of packets. IKE processes handle key negotiation, and renewals are implemented in FreeS/WAN's standalone pluto dæmon.

### Requirements and Installation

First, you will need a PC with a floppy disk drive (I have tested only 3.5" disk drives) and two network cards in it. The demands of LRP (the distribution) are minimal and do not require a powerful PC. Anything that is Intel 486-class or better with more than 8MB of RAM will do. You also will need two floppy disks. Reliable, high-density 3.5" floppy disks should do, such as promotional diskettes from AOL. I have never had any problems with generic floppy disk drives, but I have found some problems with writing the distribution to floppy disks with Imation USB U2 SuperDisk drives.

You will need to download the appropriate DUCLING.tgz/zip distribution from ftp.cinemage.com/pub and extract the contents of the archive file. If you have a static IP address, then download the static version, and if you are assigned a

dynamic IP address, you will need the distribution with a DHCP client. If you are running Windows 9x, download ducling-stat-W9x-1-0.zip or ducling-dyn-W9x-1-0.zip. Extracting the .tgz file with Winzip (www.winzip.com) will produce a file, ducling-dyn-1-0.exe or ducling-stat-1-0.exe and directory modules. The .exe file is a self-extracting image that formats a floppy disk and writes the image to that disk. Run the ducling-stat-1-0.exe or ducling-dyn-1-0.exe file and place a floppy disk into the floppy disk drive. Note that any data on the disk will be overwritten.

If you are using MS-DOS or Windows 3.1, the TSR utility FDREAD.EXE must be loaded at the DOS level first if you wish to read and write to the 1,722KB format disk. FDREAD.EXE is a freeware program from Christoph H. Hochstätter.

If you are running Linux, download ducling-dyn-1-0.tgz or ducling-stat-1-0.tgz, untar the image (the example here is for the DHCP-enabled dynamic IP address distribution):

```
tar xvfz ducling-dyn-1-0.tgz
```

and write the image file, ducling-1-0.img, to a formatted floppy using the Linux fdformat and dd commands:

```
fdformat /dev/fd0u1722
dd if=ducling-dyn-1-0.ima of=/dev/fd0u1722
```

Once the floppy disk image is created as mentioned above, you will have a bootable Linux floppy diskette.

The zipfile/directory named modules contain the required network driver modules as well as optional modules for firewall masquerading. Copy the contents of the module zipfile or directory onto a separate second MS-DOS-formatted floppy diskette for the configuration portion of this discussion (below). In Linux, format a second floppy disk by running

```
fdformat /dev/fd0
```

followed by

```
mkdosfs /dev/fd0
```

and mounting the floppy drive and copying the modules over. Read the documentation included in the README files, which will give you details on configuring your firewall/router.

If you are unable to fit all the desired packages and modules onto a single floppy diskette, you will need to examine alternative setups that use dual floppy diskettes (see the included README files with the DUCLING distribution), a

bootable CD-ROM or even a small hard disk. Refer to the on-line sources of LRP documentation for further information.

### The LRP Boot Floppies—The Surprising Truth

You may be surprised to discover that LRP uses DOS-formatted floppies. You may be even more surprised to discover that the DUCLING distribution installs itself as a 1,722KB bootable disk image. The 3.5" high-density floppy is technically a 2MB format medium, and you may see these diskettes rated as 2MB “raw” or “unformatted” capacity. The 1,440KB formatted capacity is merely the result of a conventional format that writes 80 tracks on the magnetic media with 18 sectors per track. With the appropriate tools, you can create diskettes that have 80 sectors and 24 tracks per sector, giving 1,920KB per floppy. Floppies having 1,680KB (80/21 sector/tracks per sector) are used regularly for LRP distributions and seem to have a reliable track record; 1,722KB (82/21), 1,743KB (83/21) and 1,760KB (80/22) also are reported to be in use. I have found the 1,722KB format floppy to be reliable enough for testing and have no problems to report so far.

I have created and used large-format floppies of up to 1,920KB. Extremely large-format floppies tend to be nonbootable, apparently as a result of a conflict between PC BIOSes and the nonstandard sector size on the diskette. It has been reported that large-format floppies larger than 1,680KB can suffer from floppy disk hardware dependability problems. Windows NT and Windows 2000 are reported to have reliability problems writing to large-format floppies larger than 1,680KB.

MS Windows 9x operating systems generally read standard as well as large-format floppy diskettes with no configuration changes. In Linux systems, it is often necessary to mount the floppy disk with the correct format specified, i.e., /dev/fd0u1722, where fd0u1722 specifies floppy disk device 0 (fd0) and the u1722 specifies a 1,722KB format. The standard floppy disk drive in Linux /dev/fd0 defaults to /dev/fd0u1440, the 1,440KB format.

For creating and manipulating large-format floppies, consult the LRP Boot Disk HOWTO by Paul Batozech. You'll find this, and other useful articles, in the resources listed at leaf.sourceforge.net/devel/thc. For MS Windows, I have found Gilles Vollant's WinImage (www.winimage.com) to be particularly useful and user friendly. However, it is in some ways more limited than the Linux tools, such as fdformat, mkdosfs and the more recent superformat application. The self-extracting 1,722KBps images for MS Windows discussed here were created using WinImage.

## How the LRP Distribution Loads

Before you begin to work with LRP it is useful to note how the distribution works. If you examine the bootable diskette, you will see a series of files, including ldlinux.sys, linux, syslinux.cfg, root.lrp, etc.lrp, modules.lrp and local.lrp.

The file ldlinux.sys is the bootstrap loader that loads the kernel (the file named linux) and initial root.lrp package into memory. The kernel starts and creates a RAM disk and extracts the root.lrp package. A RAM disk is a portion of memory that is allocated as a partition. In other words, the kernel creates a space in memory and treats it like a read/write disk. The kernel then mounts the boot device specified in syslinux.cfg. The remaining .lrp packages on the boot disk are extracted as specified in syslinux.cfg and loaded to the RAM disk. The .lrp packages are merely standard UNIX tarballs (tar-gzipped archives). Once the .lrp packages are installed in the directory tree on the RAM disk, the system begins a boot based on the standard Linux rc file boot hierarchy.

LRP is simply a stripped-down standard Linux kernel with loadable modules and other software contained in sets of .lrp packages. LRP is truly Linux; generally, anything that will run on a generic Linux distribution should run off the LRP diskette. Often the obstacle to extending LRP's applications and capabilities is the space constraint of a single diskette. If you require additional capabilities, for example, remote administration through ssh, a DNS server and so on, you will want to look at multidiskette, CD-ROM or even the full disk drive distributions of LRP that are available.

## Start up and Configuration of Router/Firewalling VPN

Once the bootable floppy disk is created, make sure the floppy is placed in the floppy disk drive of the machine on which you wish to run the firewall/VPN. Ensure that the BIOS is configured to boot from a floppy disk. Upon booting the firewall/VPN, you will see the LRP splash screen, messages from the Linux loader followed by a login prompt.

If you have made it this far, congratulations! You have installed an LRP distribution successfully. Now you can start to configure the firewall properties of the LRP as outlined in the bundled documentation.

Once any firewalling tweaks are completed, the VPN needs to be configured. The bundled DUCLING documentation discusses the details for configuring a subnet-to-subnet setup. This involves configuring IPSec's authentication mode (/etc/ipsec.secrets), the IPSec network configuration (/etc/ipsec.conf) as well as the firewalling rules to allow access to ports 500 (UDP), 50 and 51 (TCP).

Note that you need not necessarily require a static IP address in order to run VPN links. A “roadwarrior” configuration is described in the next section, in which the one VPN client has an undetermined static IP address. I have run VPNs between pairs of nodes with dynamically assigned IP addresses. The management of VPN nodes with DHCP-assigned IP addresses becomes tricky if both IP address assignments change frequently. The following section discusses a roadwarrior configuration using DUCLING and a Microsoft-based IPSec client.

### Interoperability Example

This example shows an MS Windows 9x/2000 client point-to-site using SSH Communications Security Sentinel 1.1 (Public Beta 3). FreeS/WAN is interoperable with a wide range of IPSec implementations. The ease of implementation and computability will vary depending on the product. Many IPSec products that support 3DES/MD5 encryption through IKE are interoperable with FreeS/WAN. However, I found that legally obtaining fully functional IPSec implementations that support strong encryption can be arduous, especially if you live outside of the United States.

Many vendors offer only limited capabilities in their freely available IPSec implementations. For example, a product may only support weak encryption (DES) or may limit VPN capabilities to transport mode only. It is important to distinguish between the two VPN modes that are offered through IPSec: transport mode and tunnel mode. Transport mode encrypts and authenticates traffic between two fixed end points. Tunnel mode is more useful for connecting subnets and allows tunneling through firewall and router parameters into different subnets. Basically, transport mode restricts traffic to point-to-point communication. Tunnel mode also allows point-to-site (point-to-subnet) or site-to-site communications. At least one vendor does not seem to allow its implementation of IPSec to run over a connection using a static IP address.

The SSH Communications Security Sentinel product (www.ipsec.com) does not seem to suffer from any of these problems, possibly due to the fact that the company is based outside of the US. I downloaded and tested the 30-day trial beta 3 release of Sentinel 1.1 and found it to be very easy to configure on a Windows 98 desktop PC. The Sentinel documentation provides configuration examples for interconnectivity with a FreeS/WAN VPN gateway.

Here is a summary of a roadwarrior configuration that allows remote users with dynamically assigned IP addresses to connect transparently to a LAN behind a firewall. You will need to open ports 50, 51 (TCP) and port 500 (UDP) to the dynamic IP address or the ISP's DHCP address range. Figure 1 shows the

basic setup. You will need to edit /etc/network.conf on the DUCLING FreeS/WAN firewall (go into lrcfg, choose 1), then 1) and set

```
eth0_IP_SPOOF=NO
```

to disable the blocking of tunneled packets. The bundled documentation contains the detailed instructions on how to do these tasks.

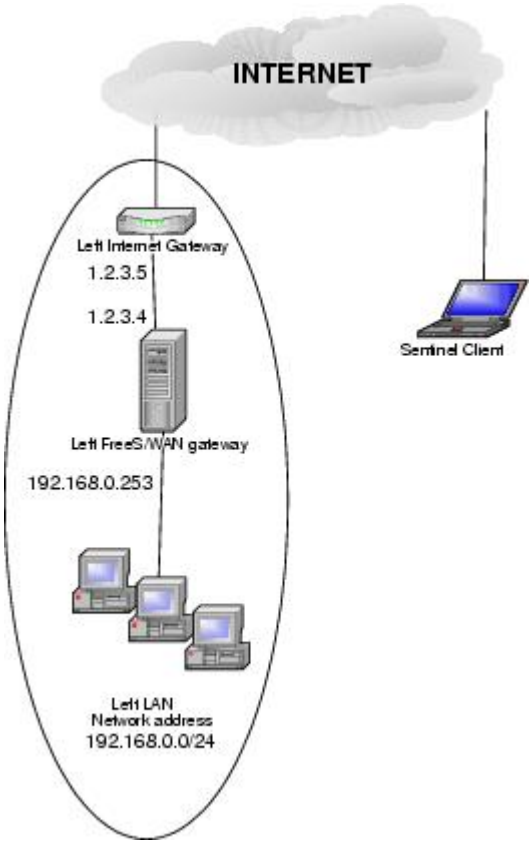



Figure 1. A Roadwarrior-to-Site Configuration

The contents of the FreeS/WAN ipsec.conf file are given in Listing 1. The corresponding ipsec.secrets file contains the entry

```
1.2.3.4 0.0.0.0: PSK "Put your roadwarrior secret
string here"
```

where the phrase in quotes is a shared-secret string. The IP address 0.0.0.0 denotes any IP address, so remember to choose a secure shared-secret string. The rightsubnet and rightnexthop parameters are left blank and imply that the connection is a point-to-subnet connection.

Listing 1. The FreeS/WAN conn Listing for the Setup Shown in Figure 1.

To set up the Sentinel IPSec service:

1. Download SSH Sentinel from www.ipsec.com and install, following the instructions.
2. Go into the Sentinel Policy Manager (Figure 2).

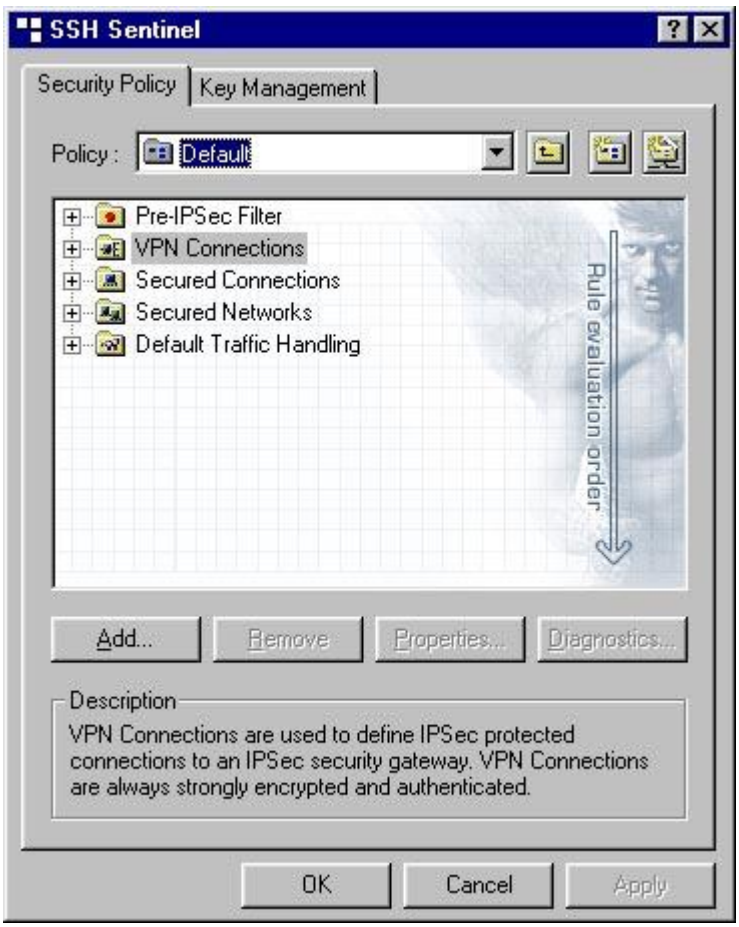



Figure 2. Sentinel Policy Manager

3. Choose the Key Management tab, Authentication Keys and select Add (Figure 3).

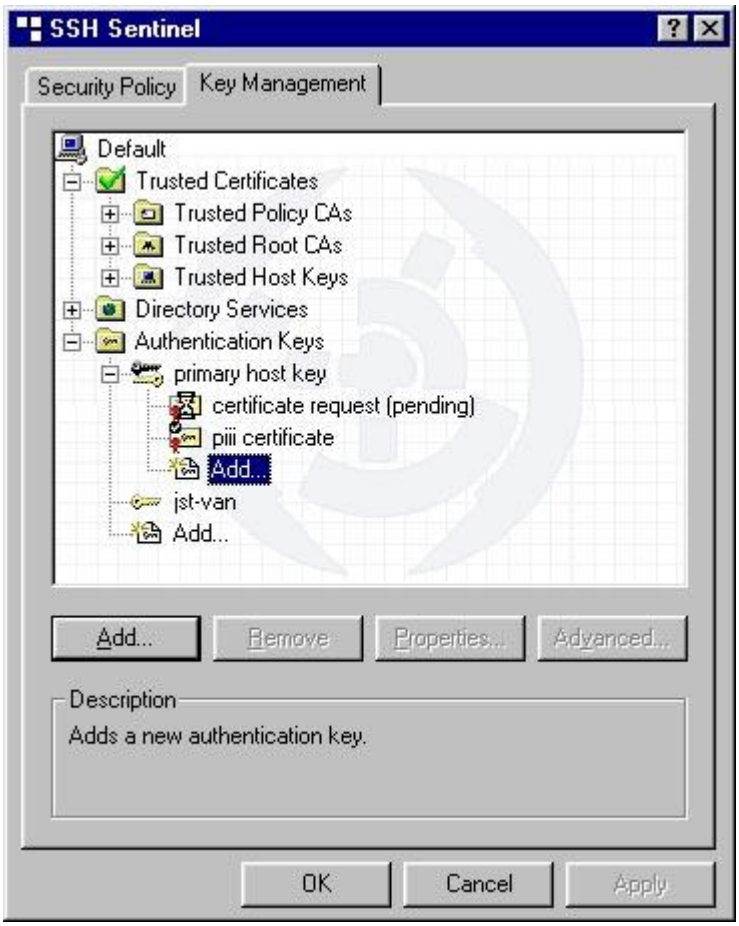



Figure 3. Adding a New Key

4. Select Create a new preshared key then Next (Figure 4).

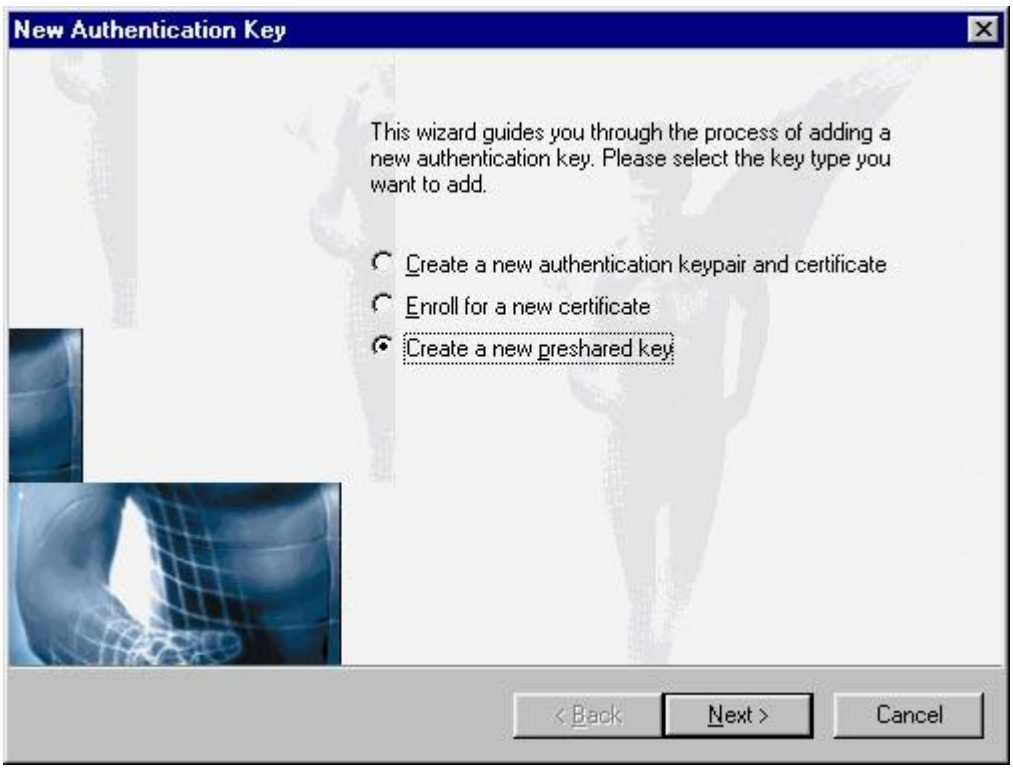



Figure 4. Configuring Preshared Key

5. Type in your preshared key. It must be identical to the shared-secret string you have inserted in /etc/ipsec.conf (without the quotes). (See Figure 5.)

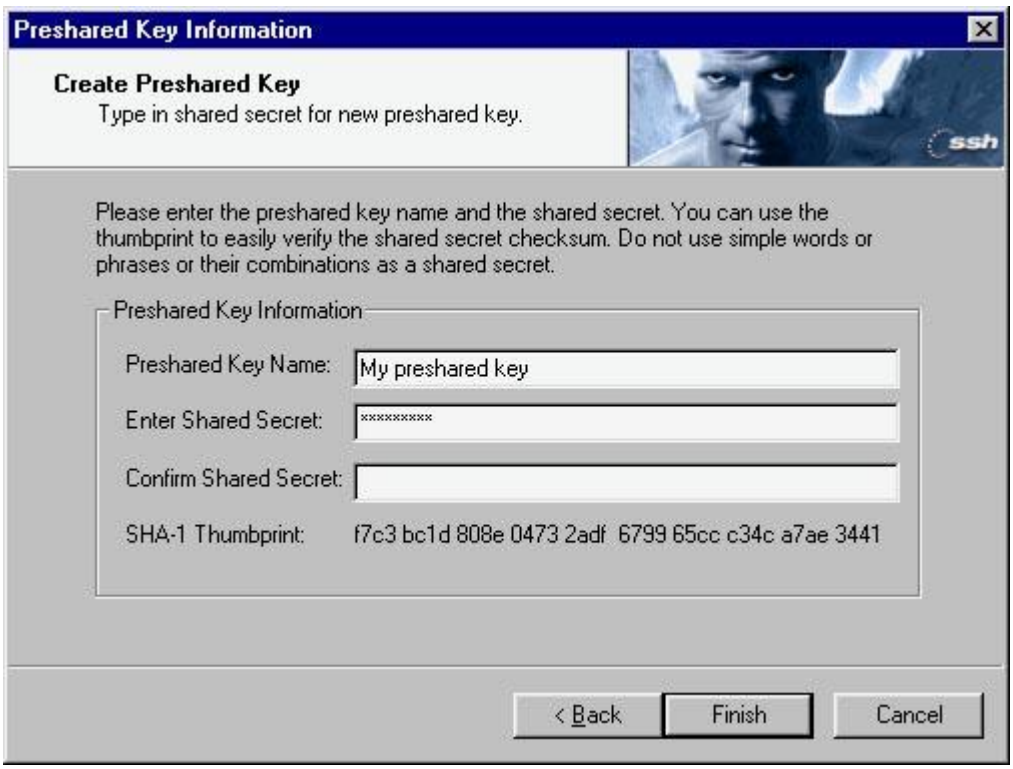



Figure 5. Typing in Shared Secret

6. Press Finish.
7. On the main console of SSH Sentinel Policy Manager, in the Security Policy pane, select VPN connections®Add.
8. Enter in the IP/hostname of the remote VPN gateway; for our example, it is 1.2.3.4, and choose the preshared secret that you created in step 5 as the Authentication key (Figure 6).


Figure 6. Entering Key and UP Information

9. Select 3DES encryption, Main Mode and MODP 1024 for IKE Mode and IKE Group, respectively. The Advanced pane generally can be left with the defaults.
10. Set the IKE SA lifetime (i.e., the interval between rekeying) to the same value as in the ipsec.conf file, typically 480 minutes (eight hours).

Save all settings and try to ping an internal node behind the firewall (try the internal interface, 192.168.x.254). You should be connected. Try running Sentinel's diagnostics to make sure you are connected. I have found that Sentinel's diagnostic mode can hang the FreeS/WAN-Windows connections sometimes. If this happens, go to the FreeS/WAN gateway and do a restart of IPSec and then bring up the various connections.

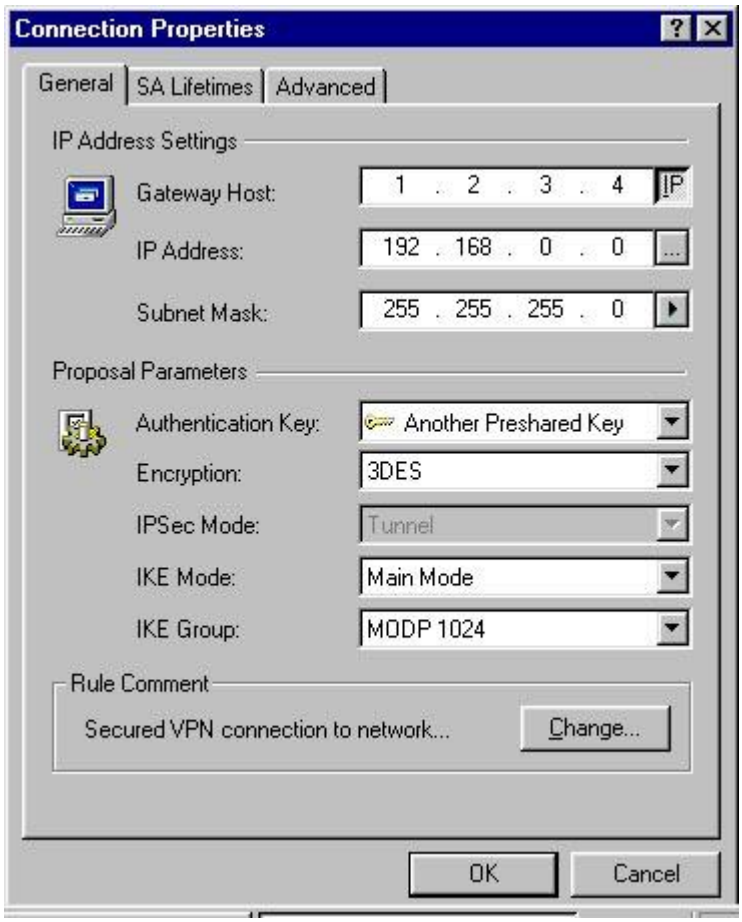



Figure 7. The VPN Connection Properties Tab

Once again, if you need to restart the connection, log in to the LRP box and type

```
#/etc/initd.d/ipsec restart
```

to restart the IPSec components.

I also found in Windows 2000 Professional (but not Windows 98) that I had to add the routing manually to the shared subnet 192.168.0.0/24 from the DOS console:

```
route ADD 192.168.0.0 MASK 255.255.255.0 1.2.3.4
```

(refer to the documentation for the Microsoft route command).

## Conclusion

This article outlines the means to implement a firewalling VPN gateway from a single 3.5" floppy diskette. With a single floppy diskette, you should be able to connect hosts and networks of various topologies securely using the Internet. The DUCLING distribution is a bare-bones distribution. Once you are convinced that a FreeS/WAN VPN can fulfill your needs, you can look at either going to a more full-featured LRP distribution or even a full-blown Linux system, implementing such things as remote access (via the secure shell, ssh, for example) or a DNS server.

Troubleshooting Resources

email: napier@napiersys.com

**Duncan Napier** runs Napier Systems Research, a Network and IT consultancy based in North Vancouver, British Columbia, Canada. He can be reached at napier@computer.org.

Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL



# Remote Linux Explained

**Richard Ferri**

Issue #93, January 2002

The basics in booting a workstation remotely, the requirements on the network boot kernel and how to configure remote Linux for various applications.

Remote Linux refers to Linux workstations or nodes that do not boot the Linux kernel from local media, but instead receive their startup instructions over a network, typically Ethernet. Due to the ease of configuring both the Linux kernel and the operating system itself, Linux is being customized to work in many disparate environments, from web serving, to cluster computing and X servers.

## Why Remote Linux?

The first question you might be asking yourself is why start Linux remotely? After all, installing Linux locally is a matter of sticking in the CD, answering a few questions and going out for a double latte while the workstation installs. This is true for the typical single-workstation installation; however, once you start to manage and install a lot of workstations, particularly in a cluster or server-farm setting, local media becomes less practical. With the advent of dense servers from companies like RLX Technologies, Inc., booting remotely becomes a necessity, as dense servers do not provide diskette or CD-ROM drives on the nodes. Dense servers expect to be brought up over the local fast Ethernet connection and administered remotely. The main advantages of remote Linux are:

- Centralized, hands-off administration: while many installations do have someone on site that can jockey CDs and diskettes 24/7, there are also many hands-off sites (sites where no one is physically present at the site for long periods of time). At these sites, when a programmer is working remotely and needs to boot a node using a special image that's on local

media, he or she is out of luck until someone is there to load the correct media.

- Dense server solution: since the trend is toward centralized remote administration for clusters and server farms, CD and diskette drives become rather anachronistic. The very presence of local media on the nodes means that the nodes must take on a certain form factor, thereby increasing the minimum size of the nodes. For higher density node packaging, CD-ROM and diskette drives are being phased out in certain segments of the industry.

- Versatility: sometimes one can fix a problem with a filesystem that prevents the node from coming up from local media. For example, you can run fsck on a local hard drive on a remotely booted machine in order to fix a filesystem problem.

- Cost and security: why pay for media you don't need? Some companies are selling workstations without hard drives and other local media that are intended for use as secure terminal servers. Secure, in this sense, means that if employees do not have access to local media on their workstations, it is more difficult to capture sensitive data.

- Helps eliminate version skew: in the case where all workstations are booted remotely from a single kernel image, it eliminates the problem of updating local media for kernel patches or enhancements. You can update the single remote kernel image once, instead of propagating the change to a set of workstation hard drives or, worse, to their local boot diskettes.

### The Remote Boot Process

The remote boot process mimics the local boot process but with a few important distinctions. From a logical perspective, without talking about the services that perform these tasks, this is basically what happens during the network boot process:

1. The node is powered up or reset and conditioned to boot from the network.
2. The node broadcasts its unique Ethernet MAC address, looking for a server.
3. The server, previously conditioned to listen for specific MAC addresses, responds with the correct IP address for the node. Alternately, the server responds to any broadcast on its physical network with IP information from a designated range of IP addresses.

4. The node receives its IP information and configures its Ethernet adaptor for TCP/IP communications.
5. The node requests a kernel over its newly configured adaptor.
6. The server responds by sending a network loader to the client, which will load the network boot kernel.
7. The network boot loader mounts the root filesystem as read-only.
8. The network loader reads the network boot kernel sent from the server into local memory and transfers control to it.
9. The kernel mounts root as read/write, mounts other filesystems and starts the init process.
10. Init brings up the customized Linux services for the node, and it comes up completely.

From this description, we see that the booting node has several dependencies on the server: a network boot kernel, a root filesystem and a method of transporting the kernel and IP information from the server to the node.

### The Basics

To get a node to boot correctly remotely, the server and client node must cooperate in several well-defined ways. There are several basic requirements for setting up nodes for remote booting:

- A well-stocked server: the server must have the proper services running (we'll talk about those later) that can provide information required by the remotely booting node.
- A method of remote power control: to start the node's boot sequence, you have to be able to power up or reset the node. You really cannot rely on the operating system on the node to be present at this time—after all, sometimes you need to reset the node because the operating system crashed.
- A network: the nodes all have to have a route, direct or indirect via a gateway, to the server. We'll only talk about Ethernet networks here.
- A network-bootable Linux kernel: the kernel can be either compressed or uncompressed, tagged or untagged. Tagged kernels, discussed later in

this article, are used by the Etherboot solution. Also, see the Etherboot web site in the Resources section for more information.

- A network loader: a method of reading the network boot kernel from the server and placing it correctly in the memory of the node.
- A filesystem: in the good old days, the filesystem was served over the network via NFS. With solutions like SYSLINUX, you actually can provide the entire filesystem via a RAM disk, sent over the network with the kernel.

Now that we have a basic idea of the flow of the remote booting process and a general list of the client's requirements for the server, let's discuss in practical terms the options for providing all these services.

### Remote Power Control

Remote power control is a small subset of the available possible remote hardware functions. Various manufacturers provide specialized hardware and software with their offerings that provide robust remote hardware control features, such as monitoring temperature, fan, power, power controls, BIOS updating, alerts and so on. The only remote hardware control features we really need in order to boot the nodes remotely are the ability to power-on and power-off, although having a reset feature is handy as well.

To force a network boot, the boot order on the workstation probably has to be modified. Typically, the boot order is something like diskette, CD-ROM and then hard drive. Most often, booting over the network is either not on the boot list at all or at the very bottom of the list, after the last of the local media. Since most workstations are shipped with some viable operating system already installed on the hard drive, a reset or power-on of the node will boot it from its local hard drive.

### Ethernet

To do true network booting, you must have an Ethernet adaptor that is PXE-compliant. PXE is the Preboot eXecution Environment, part of Intel's Wired for Management (WfM) initiative. PXE-compliant means that the Ethernet adaptor is able to load and execute a network loader program from a server prior to bringing over the kernel itself. Both the BIOS on the node and the firmware on the Ethernet adaptor must cooperate to support PXE booting. A PXE-compliant system is capable of broadcasting the adaptor MAC address, receiving a response from the server, configuring the adaptor for TCP/IP, receiving a network loader over the network and transferring control to it.

## Diskette

Although this is an article about remote Linux, and a diskette is a local media, there is a hybrid of remote booting that is so important it must be mentioned. Since many Ethernet adaptor/node BIOS combinations are not PXE-compliant, an open-source solution has sprung up: Etherboot. Etherboot provides a method of creating a boot diskette that contains a simple loader and an Ethernet device driver. Set the boot list to disk, and when the client is booted the Etherboot diskette takes over loading the driver into memory and broadcasting the MAC address, looking for a server. In the PXE case, the server is conditioned to respond with a network boot loader; in the Etherboot case, the server must respond with a tagged network boot kernel. A kernel is tagged by running the mknbi command against the kernel. (See etherboot.sourceforge.net/doc/html/mknbi.html for further information on mknbi.) The point of network booting the node is to get the kernel into local memory. Regardless of whether you choose the PXE scenario or the Etherboot scenario, the network boot path quickly coalesces—the kernel is in memory and control is passed to it.

## What Is My IP Information? (RARP, BOOTP and DHCP)

When the client boots over the network, whether using PXE or from diskette, it will broadcast its MAC address over the LAN, looking for a server that is conditioned to provide the client's IP information. This is so the client can configure its Ethernet adaptor with the correct IP information and continue the rest of the boot conversation using TCP/IP. There are several methods of providing the IP information to a broadcasting node: RARP, BOOTP and DHCP.

## RARP

RARP (Reverse Address Resolution Protocol) is the method by which an adaptor's unique 48-bit Ethernet address (its MAC) is associated with an IP address. When a client attempts to boot remotely, it will broadcast its MAC address to all workstations on the physical network. One or more of the workstations will be running the RARPD dæmon, which reads /etc/ethers to make the association between the 48-bit Ethernet address and an IP address and responds to the broadcasting client with its shiny new IP address. After receiving an IP address, the client should initiate a TFTP (Trivial File Transfer Protocol) request to get its image (more about that later). The biggest drawbacks to RARP are that it works only on the local physical network (it's not rebroadcast), and it supplies only a small bit of information, the client's IP address.

## BOOTP

BOOTP (Bootstrap Protocol) is a distinct improvement over RARP in that it provides gateway support (booting over a router) and provides far more information to the booting client. In addition to the client's IP address, BOOTP provides the address of the gateway (router), the address of the server, the subnet mask and the boot file (the bootable image for the client). Note that there can be one, and only one, IP address assigned to a particular hardware address.

The biggest drawback to BOOTP is that it assigns IP addresses to MAC addresses in a one-to-one relationship—a specific MAC address always will be assigned the same IP address. If you think about the requirements presented by a mobile office and traveling laptops, this one-to-one relationship proves to be somewhat limiting. In the mobile office scenario, users travel with their laptops and need to log in to a central server only occasionally, to pick up mail or whatever. The rest of the time, their IP address remains unassigned, which is a terrible waste of an IP address. The problem of underused IP addresses is addressed nicely by DHCP.

## DHCP

DHCP (Dynamic Host Configuration Protocol) is a logical successor to BOOTP. In fact, BOOTP is considered somewhat obsolete and has been largely replaced by DHCP. One reason DHCP has surpassed BOOTP in popularity is that DHCP supports dynamic address range assignment, while BOOTP only supports static IP assignment (a single MAC is always assigned the same IP address). The dynamic IP assignment facility of DHCP allows IP addresses to be reused among many nodes. In the mobile office scenario, a node connects to its network and broadcasts its MAC. The server, running the dhcpd dæmon, has allocated a range of IP addresses for mobile nodes and simply assigns the next IP address in the range to the broadcasting node. DHCP also manages the longevity of the IP-address assignment via a DHCP leases file.

The options to DHCP are myriad and beyond the scope of this article. For further investigation, consult *The DHCP Handbook* by Ralph Droms and Ted Lemon (Pearson Higher Education, 1999).

## Transferring the Kernel and Network Loader

After getting its IP information and configuring the adaptor for TCP/IP, the node BIOS typically requests an image over the network. This clear division of IP assignment and image serving is deliberate; it allows for IP assignment and image serving to be potentially served by different machines. TFTP (Trivial File Transfer Protocol) is just the right tool to transfer the image from server to

client, since TFTP, unlike its heavier-weight cousin FTP (File Transfer Protocol), does not require a user to log in to get a file. The primitive security built into TFTP is that, by default, TFTP only permits transfer of files from the server's /tftpboot directory. Since this security scheme is fairly well known among system administrators, only public files are put in /tftpboot. In the latest version of tftp-hpa, file-access security was added as well.

Notice that we've been talking about transferring an image—this is because the image can be either a tagged kernel (Etherboot) or a network loader (PXE). If you use Etherboot, the diskette boot method, then BOOTP or DHCP should point to a tagged kernel. If you use true PXE, then BOOTP or DHCP should point to a network loader. In the PXE case, the network loader is loaded into memory and then brings over an untagged kernel via TFTP. To use PXE, the TFTP server must support the "tsize" TFTP option (RFC 1784, RFC 2349). **tftp-hpa**, by H. Peter Anvin, supports this option and can be obtained at www.kernel.org/pub/software/network/tftp.

### Monolithic Kernel

Whether you use the two-step PXE boot process (network loader and kernel) or the one-step Etherboot process (just a kernel), eventually the kernel is read into memory on the client and control is transferred to it. You might be wondering, what are the differences between a network boot kernel and a typical kernel built to boot from the local hard drive? The first decision you have to make is whether the kernel is modular or monolithic—meaning no loadable modules. If you've ever built a Linux kernel, you're aware that features are either selected with "Y" (include the feature), "N" (do not include the feature) or "M" (load on demand). If your kernel is monolithic, you cannot have any M features.

### Kernel Flags

Several flags have to be turned on for a monolithic network boot kernel. First, if you're creating a monolithic kernel, you need to turn off modules support. In the .config file, you will see

```
#CONFIG_MODULES is not set
```

and if you're doing a monolithic kernel, you would turn it off, as in

```
CONFIG_MODULES=n
```

You must support ext2 filesystems if you intend to create them on the client or mount them from the server, as in the case of the NFS root filesystem:

```
CONFIG_EXT2_FS=y
```

If you intend to use a remote root filesystem, mounted via NFS, you will need the NFS options:

```
CONFIG_NFS_FS=y
CONFIG_ROOT_FS=y
```

Since you're going to use Ethernet to boot your remote client, you must turn on network configuration and, at least, support for the specific network interface card (NIC) on the client:

```
CONFIG_NETDEVICES=y
CONFIG_EEXPRESS_PRO100=y
```

Lastly, you must configure the ability to get your IP address via RARP, BOOTP or DHCP:

```
CONFIG_IP_PNP_RARP=y
CONFIG_IP_PNP_BOOTP=y
CONFIG_IP_PNP_DHCP=y
```

If you are supplying your root filesystem via a RAM disk, instead of over NFS (see next section), you'll have to specify the RAM filesystem options:

```
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_INITRD=y
```

### Modular Kernel

In the modular kernel, you are allowed to use M to select modules to be loaded on demand by the kernel. Since the modules are not statically bound in the kernel, there must be a place to load the modules from when they are needed. The place provided by Linux to load the kernel modules is the RAM disk. This RAM disk is used only during the boot process and only to provide the kernel modules. Another use of the RAM disk, to provide the remote root filesystem, is outside the scope of this article.

To demonstrate the use of the modular kernel, we can make support for the MINIX filesystem modular, to support a MINIX root filesystem that will be loaded from diskette. To create a modular kernel, the first thing to do is turn on modular support and rebuild the kernel:

```
CONFIG_MODULES=y
```

In our example, we're going to provide MINIX support via a dynamic module, so we'll make that option an M:

```
CONFIG_MINIX_FS=M
```

After rebuilding the modular kernel and copying it (bzImage) to /tftpboot, we're going to need a way to load it. Tools like the open-source SYSLINUX package provide the means to load the modular kernel and provide the RAM disk. A sample default pxelinux.cfg file might look like:

```
DEFAULT bzImage
APPEND vga=extended initrd=minix.gz root=/dev/fd0 ro
PROMPT 1
TIMEOUT 50
```

The configuration file says that the name of the kernel is bzImage, the RAM disk name will be minix.gz, and the client root will be loaded from diskette (/dev/fd0).

We will have to create a RAM disk that has the MINIX module (minix.o) on it, as well as insmod (the command to load the MINIX module), the console device (/dev/console) and linuxrc, the command that gets called when the RAM disk is invoked. (Refer to the initrd.txt file, written by Werner Almesberger and Hans Lerman and distributed as part of the kernel RPM, for a complete description of how RAM disks work in Linux.) To create the custom RAM disk that will mount the MINIX modular dynamically, and MINIX root filesystem from diskette, you must create a file and zero it out using dd:

```
dd if=/dev/zero of=minixroot bs=1k count=4096
```

Next, associate the file with a loopback device, /dev/loop0:

```
losetup /dev/loop0 minixroot
```

Create an ext2 filesystem:

```
mkfs.ext2 /dev/loop0
```

Mount the device over a convenient mountpoint, say /mnt:

```
mount /dev/loop0 /mnt
```

Create some favorite directories:

```
mkdir /mnt/dev /mnt/lib /mnt/sbin
```

Create the console device:

```
mkdev /mnt/dev/console c 5 1
```

Copy the minix.o module into /lib, and sash and insmod into /sbin:

```
cp /usr/src/linux/fs/minix/minix.o /mnt/lib/minix.o
cp /sbin/sash /mnt/sbin/sash
cp /sbin/insmod /mnt/sbin/insmod
```

Find out which libraries insmod needs:

```
ldd /sbin/insmod
```

and copy them to /mnt/lib:

```
cp /lib/libc.so.6 /mnt/lib
cp /lib/ld-linux.so.2 /mnt/lib
```

Then create a /mnt/linuxrc file that loads the minix module:

```
#!/sbin/sash
/sbin/insmod /lib/minix.o
```

Make linuxrc executable:

```
chmod 777 linuxrc
```

Umount /mnt and detach the loopback device:

```
umount /mnt
losetup -d /dev/loop0
```

**gzip** the minixroot file, and you're ready to boot the client:

```
gzip minixroot
```

This example will get you as far as the kernel trying to load its root filesystem from /dev/fd0. To fully test out the example, you'll have to create a MINIX filesystem on the diskette, mount it and copy over at least init and sh, and the libraries they need, and create a console device.

### Remote NFS Root

As we all know, life without a root filesystem (/) is meaningless. When booting locally, the root FS is almost always found on the local hard drive. Where does root come from on a network-booted client? To provide root you have two choices: remotely mounted root over NFS from the server or by a RAM disk. If you provide root via NFS, by default your kernel looks for root in /tftpboot/*ip*, where *ip* is the IP address of your client. This requires starting NFS on the server and exporting /tftpboot (or /tftpboot/*ip* for each node). To get the client node to boot to the login prompt, there are several requirements on the root filesystem, including the init and shell binaries; devices, at least the console device; and any dynamically loaded libraries the init and shell binaries might depend on.

A quick-and-dirty method of populating a remote root filesystem would be to copy init, sh, the necessary libraries and a console device, as in:

```
cp /sbin/init /tftpboot/192.168.64.1/sbin/init
cp /bin/sh /tftpboot/192.168.64.1/bin/sh
```

To determine the dynamically loaded libraries for init, use the ldd command:

```
ldd /sbin/init
ldd /bin/sh
```

and then copy the libraries listed by the ldd commands to /tftpboot/*ip*/lib. To make the devices, there is a handy MAKEDEV command, part of the MAKEDEV package:

```
/dev/MAKEDEV -d /tftpboot/129.168.64.1/dev console
```

If you have your other services up and running on the server correctly, when you force a network boot on the client, it will then run the init script from its remote root, using the console provided in its remote root, bring up a shell and prompt for a runlevel (since there is no /etc/inittab file in remote root). Enter s for single-user mode, and just like that, your client is up and running to the shell prompt.

A special shell is available, called sash (for standalone shell) that is extremely useful in the remote environment. This is because sash has no dynamically loaded libraries and provides some standard built-in commands that manipulate filesystems (mount, umount, sync), change file permissions (chmod, chgrp, chown) and archive (ar, tar), among other things. Instead of starting sh, for example, you can copy /sbin/sash to /tftpboot/*ip*/sbin/sash, and the kernel will bring up the standalone shell instead. You also might want to provide your own rudimentary inittab file to run sash on startup, as in:

```
id:1:initdefault:
1:1:respawn:/sbin/sash
```

## Conclusion

In this article we've explored a few of the services and methods used to boot Linux remotely. Remote Linux is extremely fertile ground for continuing research. As networks become faster and can support greater numbers of remote clients, and as clusters become larger and have greater dependency on centralized administration, remote Linux techniques will play an even greater role in the industry. With the advent of dense server technology, remote Linux has become not just a convenience but a necessity.

## Acknowledgements

I gratefully acknowledge the research of Vasilios Hoffman from Wesleyan. “V”, as he likes to be called, demonstrated the use of loopback devices in creating RAM disks and how to create modular network bootable kernels correctly. V is simply a wealth of Linux information.

Resources

email: rcferri@us.ibm.com

**Richard Ferri** is a senior programmer in IBM's Linux Technology Center, where he works on open-source Linux clustering projects such as LUI (oss.software.ibm.com/lui) and OSCAR (www.openclustergroup.org). He has a BA in English from Georgetown University and now lives in upstate New York with his wife, Pat, three teen-aged sons and three dogs of suspect lineage.

Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL



# VNC, Transparently

**Jeremy Impson**

Issue #93, January 2002

Secure, transparent and ubiquitous desktops with VNC and OpenSSH, Part 1 of 2.

This two-part series presents a novel way to set up a VNC-based X Window System desktop for your Linux system. By the end of this two-part series, you'll have a configuration that allows users to log in to their X-Window desktop (running GNOME, KDE or other preferred window manager environment) via a display manager (like GDM, KDM or XDM). More importantly, the user will have secure access to the same desktop in the same state from the workstation console and anywhere else on a network.

Typically, a workstation system runs a display manager. In this article we refer to such applications as XDM, GDM (GNOME Display Manager) or KDM (KDE Display Manager) generically as display managers. A display manager provides a graphical login prompt for the user. When a user logs in, the display manager starts the appropriate window manager (such as fvwm2, GNOME or KDE). From the window manager, the user can run whichever applications he or she wishes. When the user logs out, applications are closed, the window manager exits and the display manager reappears, ready to allow another user to log in. If the same user logs in again, the display manager starts the window manager anew, and all the applications must be restarted. This is how traditional X Window System desktops work. We refer to such a desktop session as an X desktop. We also say that when a user is using the keyboard and monitor of the workstation, he or she is logging in to the console. This is as opposed to connecting via the network.

Figure 1. A Display Manager

In the *Linux Journal* article titled “Virtual Network Computing” by Choong Ng [available at www.linuxjournal.com ], we learned how to set up VNC in order to allow stateful access to a desktop from any computer on the network. By stateful, I mean that when a user is not connected to the desktop, the desktop does not terminate but remains waiting for a user to reconnect. When the user connects to the VNC server using a VNC client, every window is in the place where it was last left, every application is still in the same state as when last used, and every opened file remains opened in the same position. The nature of the VNC server, which controls the window manager and the applications, permits this.

Therefore, any computer on the network can run a VNC client (such as vncviewer) to connect to the workstation and display the desktop. We even could run the VNC client on the workstation on which the VNC server is running. We refer to such desktop sessions as VNC desktops, and we refer to the workstation where we run the VNC server (and its window manager) as the VNC workstation.

There is one problem with VNC desktops. Suppose you want to log in to the console of the VNC workstation. Your VNC desktop is running on this workstation as well, the same one you connect to from many different computers on the network. You want to continue to have access to the VNC

desktop via the network. At the same time, when you log on to the console via a display manager, you want to see the same desktop you see when you connect via VNC. But if you log in to the workstation via the display manager, it will start a new window manager. Basically, you have started a new X desktop, one which is independent of the VNC desktop, already running on this workstation.

If you want to connect to the VNC desktop, you must run a VNC client, such as vncviewer. This is awkward due to the fact that one window of the X-based desktop is itself another desktop (the VNC desktop). Keeping track of the many levels of redirection can be troublesome. Besides being confusing, due to ambiguity as to what desktop the user is actually using, it also is inefficient as it requires two window managers to run concurrently, when only one is needed.

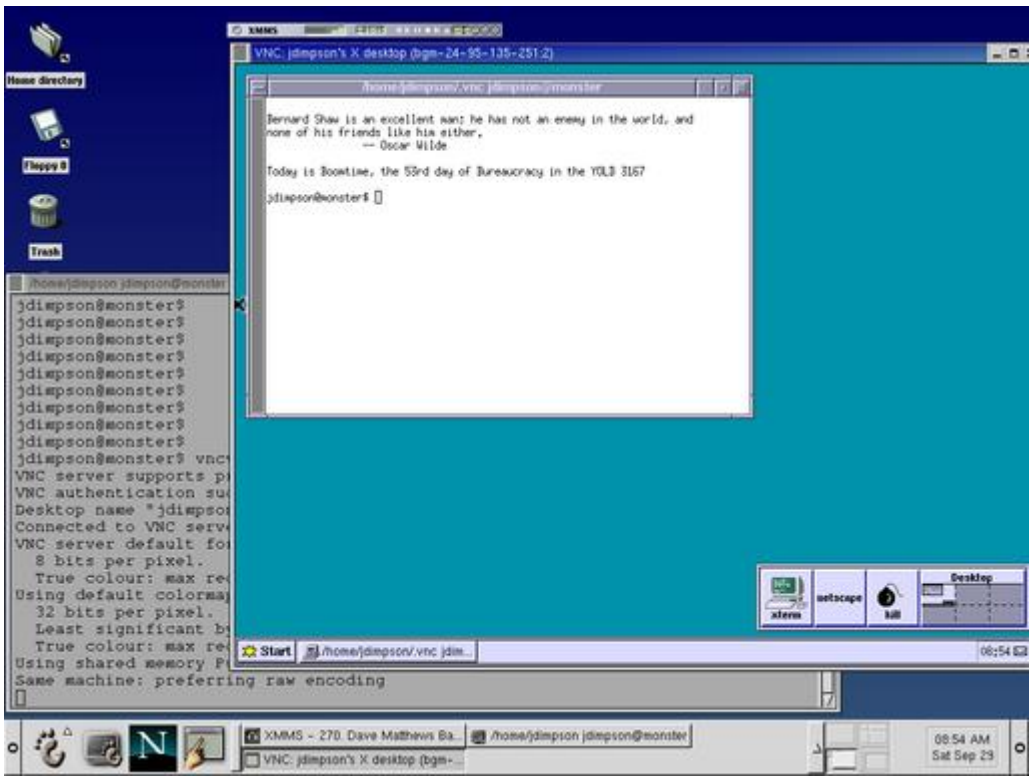


Figure 2. Screenshot of an X Desktop Running vncviewer, Displaying a VNC Desktop

This article explains how to configure an X server, display manager and a VNC server so that the desktop one sees when logging in to the display manager is the VNC desktop, with no second window manager and with all files and applications in the same state as they were last left.

## Prerequisites

The scheme we discuss can work on any Linux distribution. It requires a working X server, a display manager and VNC. I checked for these packages with this command:

```
rpm -q XFree86 vnc XFree86-xdm kdebase gdm
```

It is only necessary to have either XFree86-xdm kdebase or gdm installed. I should note that all the configuration file locations discussed in this article are as shipped with Red Hat 7.1. It is possible to configure any Linux system to allow transparent VNC desktops, but you may have to download software or locate configuration files if they are in different locations.

Whatever display manager you prefer, it should start at boot time. This is usually accomplished with a line in /etc/inittab similar to this:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

**prefdm** is usually a copy of a link to whatever display manager you prefer. X and your preferred display manager must be up and running.

### Configuring a VNC Server

The VNC server also must be running, and it must be configured to run your preferred window manager. This is accomplished by editing the file $HOME/.vnc/xstartup to call your preferred window manager. Use **startkde &** for KDE, **gnome-session &** for GNOME or **fvwm2 &** for FVWM2. Also, make sure you have run **vncpasswd** in $HOME/.vnc/passwd to create the password file.

Red Hat 7.1 provides an easy way to start up the VNC desktop at boot time. Use **linuxconf** to set the vncserver boot script (in /etc/init.d/vncserver) to come up at boot. The default bootscript, however, doesn't quite give me the flexibility that I'd prefer. Edit /etc/init.d/vncserver, looking for the line that says

```
"su - ${display##*:} -c \"cd && [ -f .vnc/passwd ]
&& vncserver :${display%%:*}\""
```

Change it to look like this:

```
"su - ${display##*:} -c \"cd && [ -f .vnc/passwd ]
&& vncserver ${ARGS} :${display%%:*}\""
```

Then edit /etc/sysconfig/vncservers so that it looks like this:

```
# The VNCSERVERS variable is a list of
# display:user pairs.
#
# Uncomment the line below to start a VNC server on
# display :1 as my 'myusername' (adjust this to your
# own). You will also need to set a VNC password;
# run 'man vncpasswd' to see how to do that.
#
# DO NOT RUN THIS SERVICE if your local area network
# is untrusted! For a secure way of using VNC, see
# <URL:www.uk.research.att.com/vnc/sshvnc.html>.
VNCSERVERS="1:jdimpson"
ARGS="-geometry 1024x768 -alwaysshared "
```

Change the value “1024x768” in ARGS to represent the size of your actual X desktop. Add any other VNC server arguments that you wish to this ARGS

variable. Change jdimpson in VNCSERVERS to whatever user you wish to run the VNC Desktop. The value “1” in VNCSERVERS makes the VNC server run as display 1. You can have additional desktops come up like this:

```
VNCSERVERS="1:jdimpson 2:phred 3:sysadmin"
```

On a Red Hat system, make sure the VNC server is running by executing this:

```
/etc/init.d/vncserver start
```

At this point, you can connect to the VNC Desktop using any VNC client.

### Configuring the Display Manager

On my Red Hat 7.1 system, I created a file called $HOME/.xsession. This file says which program, usually a window manager, should be run when I log in through a display manager. When logging in, the display manager checks for the existence of this file. If it exists, the display manager will then run the program listed in the file. The display manager considers this file to contain the command to start the user's desired window manager. Instead of running a window manager like GNOME or KDE, however, we'll run the VNC client. Edit $HOME/.xsession so that it looks like this:

```
exec vncviewer -passwd $HOME/.vnc/passwd
-fullscreen localhost:1
```

If you are using another Linux distribution, it is highly likely that the same mechanism will work for you, but you should double-check. One quick way to check is to put the line

```
exec fvwm2
```

in this file (assuming fvwm2 is loaded on your system). When logging in to the display manager, if fvwm2 starts up, then you have success. If not, you'll have to dive into the documentation for your system or the configuration file; try understanding what /etc/X11/xdm/Xsession does.

### Login

Log in to the display manager login window. You will be presented with your preferred desktop.

When you log in to the server using the display manager, it will be replaced by your chosen window manager running under the VNC server. If you have another computer on the same network, trying using the VNC client to connect to your server. You should see *two* copies of your desktop. When you move a window using one computer, you will see the window move on the other, too.

If, after logging into the display manager, the screen flickers and the display manager login banner returns, then something went wrong. Make sure vncserver is running, and make sure the .xsession file is correct.

Notice that in this setup you should not use any logout feature that may be part of your window manager. Doing so will end the VNC desktop, which is probably not what you want to do. Instead, just press Ctrl-Alt-Backspace to kill the X server. You'll see the display manager return to the screen. If you log back in, you'll be right where you left off. So you can turn the console over to someone else but not lose your desktop state.

## What's Happening?

When the server boots, it will run the VNC server for each user in the /etc/sysconfig/vncservers file. As it starts up, the VNC server reads the .xsession file in your home directory and will use it to run your preferred window manager. Then the VNC server will simply sit in RAM, waiting for a connection (either a local connection or one from over the network).

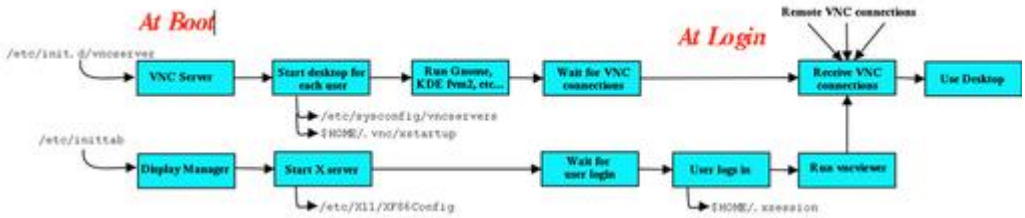



Figure 3. The Whole Process

The display manager also starts up at boot time, which presents you with a graphical login banner.

A user who does not have the VNC server configured, and who does not have the proper .xsession file in his or her home directory, will get a normal X desktop when logging in to the display manager. A user with all of these things configured will get the VNC desktop and also will be able to access the VNC desktop from anywhere on the network. Doing so securely, however, is a tale for the next article.

## Drawbacks and Other Options

The setup I've described here has many advantages. For me, the primary advantage is the ability to access my desktop's state from any computer at work, which is nice if I'm on the other side of the site with a half hour between meetings, and I want to check my e-mail or my calendar.

A major drawback of using VNC as your default desktop is that you lose graphical performance. For example, playing movies on your VNC desktop is slow, due to long redraw delays. Most fast-action games are slow as well. Also, when the VNC server runs, it acts like an X server to all the applications that you run, but it is not accelerated in any way. Even if your true X server is accelerated, you won't be able to take advantage of it. (You can log out, log in as a different user, play the movie or the game, log out when finished, then log in again as your normal user. Your original desktop will be sitting there just like you left it.)

This configuration doesn't scale well for multiple users. You could define many VNC sessions in the /etc/sysconfig/vncservers file to start up at boot. But all of these VNC desktops will be idle until they are used. And for each VNC desktop, a VNC server is run, a window manager is run and, in the case of GNOME or KDE, many auxiliary applications are run. All of these take up RAM, and they may compete with each other for resources like the sound card. Similar commercial solutions like Citrix MetaFrame and Microsoft Terminal Server call for seriously powerful computers to support many users, and I'd guess this solution would work just as well on such hardware.

One alternative solution is to use XDMCP, which is how the traditional X world provides remote X access (à la X terminals). But in doing so you lose statefulness because the desktop is restarted every time a connection is made to the server, and you lose the ability to share the same desktop both remotely and locally. See the documentation for your display manager (xdm, gdm or kdm) or www.linuxdoc.org/HOWTO/XDMCP-HOWTO for more information.

Another solution is to run VNC out of inetd/xinetd, using the -inetd option. Doing so, however, causes the VNC server to start anew for every connection, disallows multiple connections to one desktop and shuts down after the original connection exits. So it loses statefulness and the ability to share the desktop both remotely and locally. See the VNC server documentation for more information.

Another option is x0rfbserver. This is an application that you run in your normal X desktop that will relay the contents of the desktop to a VNC client. It works well and will let you take full advantage of any accelerated video cards that your X server supports. It also is less RAM-demanding than running an X server plus a VNC server (it only requires an X server, besides the x0rfbserver itself, which is fairly small). But it requires that you always keep your X desktop running on the console, so it will not scale for multiple users. See www.hexonet.de/software.en for more information.

email: jdimpson@acm.org

**Jeremy D. Impson** is a senior associate research scientist at Lockheed Martin Systems Integration in Owego, New York. There he's a member of The Center for Mobile Communications and Nomadic Computing, where he uses open-source software to develop mobile computing systems. He can be reached at jeremy.impson@lmco.com.

Archive Index Issue Table of Contents

Advanced search

# Meeting with Costa Rica's Minister of Science and Technology

**Phil Hughes**

Issue #93, January 2002

A discussion of connectivity, Linux and the future of open source in Costa Rica.

I just had the pleasure of meeting with Guy F. de Téramond, who is the Minister of Science and Technology of Costa Rica. Don Guy, as he is referred to by his staff and others, is a serious Linux and free-software advocate. As ministers are appointed by the president to their cabinet-level positions, this puts a pro-Linux person very high up in the government.

Guy F. de Téramond

I originally met him in June 2001, at the Costa Rica Linux User's Group (GULCR). At that time he really impressed me with his interest in Linux and with what I saw as a serious interest in bringing good internet connectivity to the general public. This meeting was a chance to fill in the blanks of my knowledge of his interests and commitments.

The meeting was in his office at the Ministry. When I arrived he was experiencing a computer problem: he had built a new kernel for his Mandrake-based Toshiba laptop and deleted the old kernel, before he had made sure the new one was in place and worked. He chalked this up to a learning experience and, after our interview, went back to doing an update from the Mandrake 8.1 CDs.

Guy is actually a research physicist who received his degrees in Paris between 1968 and 1977. He spent a year at Harvard and a year at Stanford. He is also a Guggenheim Fellow (1986), received a Fulbright Research Award (1983), National Prize Clodomiro Picado Twight (1979) and the Medal of the Association of Space Explorers (1997).

He has served as a full professor of Physics at the University of Costa Rica from 1982 until 2000, when he was appointed to his current position. He is also a member of the American Physical Society and a founding member of the National Academy of Sciences in Costa Rica. I could go on, but I think you have the general idea.

He was responsible for connecting Costa Rica to BITNET in 1990 and to the Internet in 1993. When asked “Why BITNET?”, he explained that at the time the Internet was primarily a US phenomenon for computer scientists, while BITNET was used for collaboration in other fields.

When he returned to Costa Rica he was convinced of the potential that this connectivity could bring to the country and was responsible for the creation of CRNet, a fiber router-based backbone linking all major academic and research institutions in Costa Rica. Going beyond the borders, he was involved in the interconnection of Nicaragua, Panama, Jamaica, Honduras and Guatemala to the Internet.

I asked him about his passions, and he said theoretical physics is first and computers and the Internet are second. He smiled and declined to comment on his third.

He saw CRNet as something he would create to get the needed connectivity and then go back to his work in theoretical physics. His knowledge and experience with computing got him tied up in this area more than he had planned. What he has done and continues to do, however, is a huge benefit to Costa Rica.

### Guy's Introduction to UNIX

The first UNIX system in Costa Rica was an IBM RISC system. Back in 1991, Guy was in the first group of people that attended training on that system. Another

attendee was Mario Guerra, who has become a UNIX expert and is also currently working at MICIT.

Guy soon became involved in getting other Caribbean nations connected to the Internet. This usually meant traveling to other countries, from Nicaragua to Jamaica, and helping them set up. In each case, the goal was to make them self-sufficient by teaching them what they needed to know. “We were working together with the people because if you go and make the connections, and nobody learns anything, they will become dependent on you, and that's not the idea”, he said. “That's the idea for a private company.”

“[With CRNet] we connected 24 universities and ten government offices”, he said. “We always had the strategy to work with the computer people because they will learn faster. Thus, even though there were very few people at CRNet, we [did] a lot.”

This work continued and included more efforts to increase the bandwidth available to Costa Rica, including a PanAmSat link and a downlink-only link. Mario worked on proxying to decrease the need for incoming bandwidth.

### His Current Project

When he came to the Ministry in 2000, Guy wanted to continue this connectivity project but at the national level—that is, bring broadband connectivity to anyone who wanted it.

Costa Rica received a $1.2 million grant to start a DSL pilot project. This project offers DSL connectivity in five of Costa Rica's 240 phone districts. “We are just using the infrastructure that is there; just putting the logical elements on top of the fiber and then using the copper line”, he said. “DSL is fantastic technology because you are using what is there.”

“Now we are going into the second phase”, he added. “The second phase has two crucial parts. We are going to deploy 100,000 DSL lines in all 240 of the nation's phone districts.”

The first country in the world in per capita broadband connectivity (cable and DSL) is South Korea, number two is Canada and three is the US. With 100,000 lines, which is 10% of the copper lines in the country (and 2.5% of the total population), this effort will move Costa Rica up to number three.

The total cost of this effort will be about $60,000,000. This investment, while a huge amount for Costa Rica, is relatively small because DSL takes advantage of the fiber and copper infrastructure already in place. Also, deploying DSL helps local telephone districts because it substantially decreases the load on the

switched lines by getting all the long-time internet connections moved off the switched network. “We are paying a lot of attention to scalable technology—not frame relay, ATM and the like”, he said.

Guy pointed out that this universal connectivity is happening because of the government monopoly on telecommunications service. ICE, the telephone company, is a government agency and adding this internet connectivity through DSL makes perfect sense. If, on the other hand, a private company were to offer internet connectivity, it wouldn't be universal because connectivity to areas of low population would not be profitable.

This is similar to what happened in the United States with the REA, which brought electricity to virtually everyone. Today, 97% of the households in Costa Rica have electricity (100% from natural sources), and the modern equivalent of the electricity effort is internet connectivity.

As Guy said, “You can do so much work from your office in your house and spend more time with your family if you have good communications.” In addition, the increasing numbers of Costa Ricans moving to population centers has stressed the transportation infrastructure and created pollution in, for example, San José. Thus, the ability to telecommute is important to the nation as a whole.

In addition to this fiber/copper infrastructure, they are experimenting with communications over the power line. The electrical distribution also is done by ICE so, for example, the existing fiber run along the power grid for monitoring will be made available as a backup for the telecommunications fiber links.

To support all this connectivity, there needs to be increased bandwidth to the Internet outside of Costa Rica as well. That brought the discussion to Arcos. Arcos is a ring-based system in the Caribbean that covers Mexico, the other countries of Central America, the Caribbean islands and Miami. It has a bandwidth of about a terabyte/second, and its structure is such that all the repeaters will be located on land, making an upgrade much less expensive than typical sea-based repeaters.

Arcos should be in operation by the time you read this article. Guy explains they decided to go with Arcos because it offers much less expensive bandwidth than did satellite links.

He also stressed that the project is to connect everyone: individuals, schools health centers and such, as well as banks, commerce and high-tech businesses.

There already has been a hearing on rates, and all the comments have been in the direction of reducing them further. (The proposal was $30/month for 64k DSL, $40/month for 128k and so forth, including the line and the ISP.) As ICE is a public agency, the rates will reflect cost of service rather than the need to show a profit.

The engineering part of the project has just been completed. The design is scalable and offers, for example, easy ways to implement security. They have been careful to follow ITU standards, so there will be no dispute over what they have specified.

The request for bids are about to go out for the additional equipment, and initial funding has already been approved by the government. The goal is to start deployment in January 2002.

### What about Free Software?

Knowing that Guy is into Linux, we discussed the future of free software in Costa Rica.

**Phil** How does Linux play into this? Your connectivity project makes it possible for someone who lives in Quepos [a coastal city far from the capital] to decide they want to become a computer consultant. Is there something else that you have planned to get Linux into the community?

**Guy** At the University [of Costa Rica], there was a very integrated system of different technologies. At the end-user level, every user or department was free to put whatever OS or application they wanted on their systems, but most picked Windows or Mac—mostly Windows at the user level. A few crazy fellows in the Math and Engineering departments picked Linux.

In the center we have about 100 servers. They were all running Linux, except the databases, which were, at that time, running under Solaris on SPARC.

I think that Linux, in the last two or three years, has really made some progress for the end user. What we see again, if you use layers, is at the bottom you have common infrastructure and at the top you have the user. The more you go toward common infrastructure, the more it is open source, and the more you go toward the end user it is proprietary software.

In the coming years, we will see this line going up or going down. That is going to depend a lot on what the Linux and Open Source community does in developing applications for the end user.

**Phil** I agree.

**Guy** So this is a moving frontier. Of course, now almost everything is open source. You have the people who developed the TCP/IP protocols and Tim Berners-Lee with the HTML protocol. They made a fantastic contribution to humankind, then put it out as open source.

And, by the way, who is the biggest user of open source in the world? It is Microsoft, for example, with TCP/IP. They can put Microsoft Network on the Web, but it is TCP/IP.

**Phil** And, if it isn't, it doesn't talk.

**Guy** Exactly. In 1995, Microsoft protocol was NetBUI and NetBIOS, and it wasn't routable. Those are LAN protocols. At some point they made the very important decision to embrace TCP/IP. One has to agree that it is easy to satanize open source, but they are using it [themselves]. So we have this moving frontier. To go a little bit above, you have the web servers. You have Apache vs. IIS. Not only does Apache have 60% of the market, but it is free and reliable. And you have the report from The Gartner Group, for example, about the security issues, which are not a joke.

All of this to answer your question. To give you an example, here at the Ministry all of our servers run Linux. We are now extending that to many areas of the government. For myself and our engineers, we run Linux for everyday life. And because of the viruses, I can't go back.

However, I must say that I use a modified kernel because I really need to run [MS]Office. To be fair, Word and Excel are the standards. In open source we have not reached that level, yet.

**Phil** Even Linus uses Microsoft applications such as PowerPoint. They may not know how to write an operating system, but they write good applications.

**Guy** Exactly. I use this modified kernel that works beautifully, and I have my PowerPoint presentation and use Word and Excel because I don't have time to translate these files to another format. But it [Windows] is a program running under my OS, and I don't care if it has a virus or not.

I have the best of both worlds. The worlds will complement each other and that is healthy.

A country like ours has become very good in software development. It is one of our most important industries. So we must, of course, push this industry.

Let's move to the second question, what are we going to do about that? Here at the Ministry—engineers, myself, Linux fans—we will not go back. I do my scientific papers in LaTeX; everything is there.

At this point we can't make a decree to put the whole government on open source—yet. We have tasks with secretaries where they will lose their productivity. They cannot work as fast.

This will change in the future. However, this is a process. Right now, mail servers, web servers and databases are on Linux. One of our engineers can monitor every server from here. It's powerful, it's secure and it's the best environment you can find in the world right now. Today you cannot say that all the Ministries will run on Linux. We are not there yet. The day will come when the office productivity suite will be integrated into the desktop and will be considered common infrastructure, but not yet.

**Phil** So, I guess that answers the next question, which is that we, the Linux community, have to keep building better tools for the user, and the transition will happen.

**Guy** Absolutely. There is a second aspect to this. It's about education and schools. Here the pedagogical and instructional value of open source is incomparable.

You need to teach all primary and secondary school students the basic skills with Word and Excel, of course, but in the higher grades the values of open source are incomparable because the first to the last line of code is there. You know the program. It's amazing.

So imagine following the program line by line. You can change it, adapt it and make a simple version to monitor an experiment. Our industry will benefit from that because we will have very good programmers. The educational value, at a higher level, is very significant.

This is, of course, very close to the scientific model, where the results are open and free for discussion; only the best results remain. Only the best code survives.

I would like it if open source comes more into the high schools. That is something I would like to push more, but it depends on the Ministry of Education. I would like to see the bright kids changing the code and compiling and making a lot of mistakes, like I did.

**Phil** When offered the opportunity to work with the code, I have seen people get much more interested in getting involved. Where do you go from here?

**Guy** The problem is not what to do, it is the time to do it. As I said, my dream was to do frontier science from wherever you are. Of course, I am now extremely excited to follow the development of networking. This complements my career in Physics well. I think it was easy for me to go into this project because it is easy for me to have order of magnitude estimates.

Just to give you an example about why it is important to have an education in basic sciences, there was discussion of whether the backbone was to be wireless or with fiber. Someone who is trained in physics knows that in fiber the frequency is 1015. So that means information you can convey on a fiber, for all practical purposes, is infinite. With microwave, the frequency is between 106 and 108. So this discussion was irrelevant because you cannot pack more than an order of magnitude or two of bits at the frequency.

**Phil** I'm very impressed with your philosophy and, generally, with what I have seen as Costa Rica's philosophy. In the US there is so much argument about what the government should supply. Health care is a perfect example. Healthy people can be more productive. If the government can supply this infrastructure, the kind of connectivity you are talking about, for example, you're going to get more done. You are going to create people that are out there to accomplish things.

**Guy** I think health care is a very good example. Here it is universal. You are not going to die in the street.

**Phil** I had one other question about percentages of free software usage and such, but it seems you are saying the real answer is that the Linux and Open Source communities need to continue to develop user applications, and the transition will happen naturally.

**Guy** Absolutely. The progress in these last couple of years has been impressive, but of course, we need to continue to work much more.

## Conclusion

I enjoyed interviewing Don Guy, and I am very impressed with him. His interest in education and universal access is a real credit to his dedication. The fact that this is happening is a real credit to the Costa Rican government in general.

I mentioned to him that we were giving trips to Costa Rica to the winners of the first contest we had in *Embedded Linux Journal*, and he said that he hoped he could do something “official” for them. So my bottom line is that Linux and open source are alive and well in Costa Rica, and it feels like it could expand easily in the whole region. Hopefully, Costa Rica can set the example for domination by Linux in Central America.

Resources

**Phil Hughes** is the publisher of *Linux Journal* and *Embedded Linux Journal*.

Archive Index Issue Table of Contents

Advanced search

# Starting Share Files with NFS

**Olexiy Tykhomyrov**

**Denis Tonkonog**

Issue #93, January 2002

Olexiy and Denis provide an introduction to the features of network filesystem (NFS).

If you have two or more machines, you may wish to share disk space or devices, such as a CD drive, between machines. For this there is network filesystem (NFS), the easiest way of sharing files and resources across a network.

NFS is the way of working with files on a remote machine as if they are on your local one. NFS is the standard de facto solution to access the same $HOME directories from any machine on the Net. Using it also is a good way to avoid keeping duplicated information from eating large amounts of disk space.

NFS originally was introduced by engineers at Sun Microsystems in 1985. Quite ancient, but still good, NFS continues to experience improvements. Sun is working on producing an NFS version 4 implementation for Linux that is now in the development stage.

The standard version of NFS for Linux is 3. This article deals with this version. From an ordinary user's point of view, there are only few differences between versions.

Installing NFS both on the client and the server sides is not so hard. We show some basic NFS features and explain their use. Many commands have to be executed as root, so think 37 times before pressing Enter.

But first, how NFS works. NFS is the most-known service using remote procedure call (RPC). As an example, let's say a server machine, named tiger, keeps all $HOME files in the directory /home. From your local machine, named blackcat, you would issue the command:

```
mount -t nfs tiger:/home /home
```

According to this command, mount will try to connect to the rpc.mountd dæmon on server tiger via RPC. The server will check if the request is permitted to mount /home from blackcat and return a file handle that in turn may be used to serve all requests to files below /home. If the request is not permitted, we will see the corresponding diagnostic message. Note that the colon in the command indicates the filesystem is remote, so -t nfs may be omitted. Let's imagine the file handle was returned successfully, and we are dealing with a kind tiger.

When a user on blackcat tries to access a file on NFS, the kernel makes an RPC call to the NFS dæmon, usually rpc.nfsd, passing as parameters the name of the file, user and group ID (uid and gid). Thus the server, tiger, can prevent unauthorized access before sending the file handle. The process of mounting is shown in Figure 1.

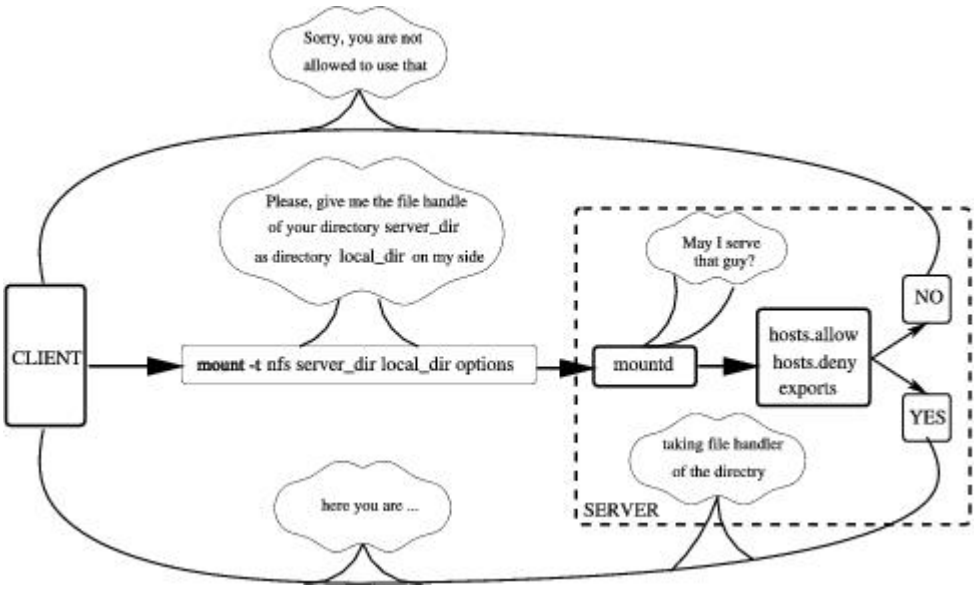



Figure 1. Mounting Process

## Setting up the NFS Server

If you want to have your Linux machine acting as the NFS server, you have to run the rpc.mountd program there. This program helps the stage of mounting and passes mount-NFS requests to the rpc.nfsd program, which does almost all work and acts as the NFS server. RPC protocol also defines using performance statistics from the kernel on the serving machine, which in turn helps in working with rpc.lockd and rpc.rquotad. Since kernel 2.2.18, rpc.lockd is evoked by rpc.nfsd on request and thus may not be started manually. The rpc.rquotad program has to be started in order for quota service to work on NFS.

As you remember, we mentioned RPC. This means that for NFS to connect we do not need to change /etc/inetd.config for the program that rules the internet

super server, inetd. NFS uses another program, portmapper, which shows the way to find all NFS services on the system.

The very first guard to protect your machine from everlasting net requests uses a pair of files, /etc/hosts.allow and /etc/hosts.deny, to allow the requests to be processed or rejected. This guard recognizes two things: the type of request and the host that sends it. Then it has a look at the file hosts.allow to find those parameters. If found, the request will pass. If not, the guard sees /etc/hosts.deny, and the inquiry is rejected if the pair matches. If the pair does not match either /etc/hosts.allow or /etc/hosts.deny, the request will be passed.

If you are setting the server, say, for an educational establishment, you have to pay more attention to security. In any such establishment there is, for instance, “the fifth column”, usually from students: one part is sniffing your passwords and other sensitive information, the second one is sending “I love you” e-mails, the third gifts you with Trojan horses. NFS, especially its old releases, has a set of features with potentially low-security levels. So, better to set up these files correctly.

To find out more about these files, consult hosts_access in man page section five. But, it is usually enough to forbid access to portmapper and other NFS requests for all machines, and allow demands for particular hosts or subnets. So, in the file /etc/hosts.deny you need to put these rows:

```
portmap:ALL
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL
```

After this action, nobody can access your NFS server, and of course this is not what we need, so let us also change the /etc/hosts.allow file. The common format for the lines in the file look like this:

```
daemon list : ... user pattern@host pattern
```

For example, in order to allow using NFS for machines blackcat and tomcat with IP addresses 192.168.16.13 and 192.168.16.24, respectively, we have to add the corresponding lines in /etc/hosts.allow file for all services that we disabled:

```
portmap : 192.168.16.13  192.168.16.24
lockd   : 192.168.16.13  192.168.16.24
mountd  : 192.168.16.13  192.168.16.24
rquotad : 192.168.16.13  192.168.16.24
statd   : 192.168.16.13  192.168.16.24
```

If you are using an NFS connection with a set of clients that usually occupies a continuing subnet and so have the corresponding addresses in the range, the row with allowed hosts may be too long. To simplify, there is another way of pointing to such a set of clients: in this case address/netmask is used, for it

makes the files shorter and more readable. For example, 192.168.16.0/255.255.255 matches all IP addresses in the range 192.168.16.0 to 192.168.16.255.

To this point, we mentioned only common features that apply to many net services. The config file for NFS service, /etc/exports, is more specific. On those two machines of the cat family, blackcat and tomcat, mount /home and /usr/doc directories via NFS:

```
/home    192.168.16.11(rw,root squash)
192.168.16.24(rw)
/usr/doc 192.168.16.11(ro,root squash)
192.168.16.24(ro)
```

The format is clean; in the left field we put the name of the directory to export and then enumerate pairs host(*permissions*). Note: there is no space after the IP address. In the example above we may write and read in /home, and only read /usr/doc and subdirectories. The parameter root_squash is needed to protect files from client root access. This means that a user (say, a student), after having successfully cracked the client root password, may become root but still cannot access or change files that only root on the server can. This is good, and the option is set by default. The word squash means that the root user will have the same access as user nobody.

If we look at the situation with root carefully, we also will deduce that we should do something with user and group identification numbers: uid and gid. If a user has different uid and gid on a client and on the server, the user may not access fully his or her own files, but may access files belonging to another user.

To avoid this situation, you may set the file to indicate static mapping for uid/gid. In the example below, the option points to the file /etc/nfs/home_map.cats and may look like this:

```
map_static=/etc/nfs/home map.cats
```

The file for mapping, home_map.cats may look like this:

```
# Mapping for cat's machines:
#    server    client
uid   0-60       -    # switch to nobody
uid   61-80     41    # map 61-80 into 41-60
```

The process of communication between dæmons both on the server and client is shown in Figure 2.

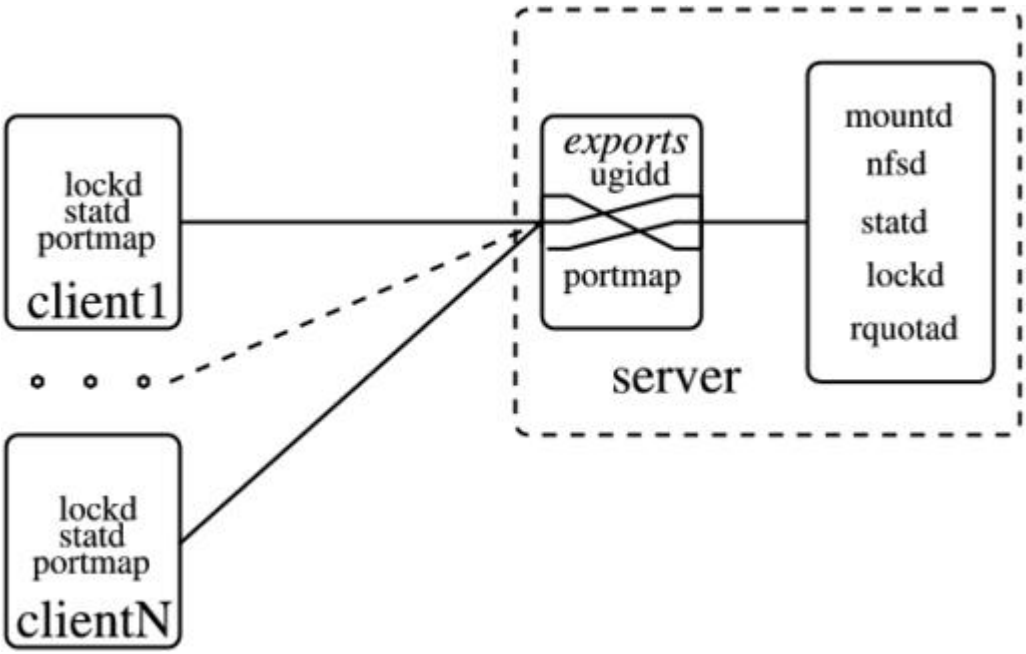


Figure 2. Communication between Dæmons

If you have an NIS system running, the NFS server may use it to consult on uid/gid. In this case you should use another option, map_nis=kingdom.cat. This solution is more clever; the NFS server will obtain the corresponding information from the NIS domain kingdom.cat.

To allow mount process for a group of trusted users described in your NIS system, you may use the parameter @group. If you put, say @catowners, only the host part of all members of the group catowners is extracted to add to the access list.

Having those files correctly installed, you should then start the corresponding programs. Because NFS uses portmapper, portmapper should be run first. In recent distributions of Linux, portmap or rpc.portmap starts while the system is booting. You may check the current status of the program by issuing the following two commands in one line:

```
ps axu | egrep portmap
```

If the portmap is running, you will see something like this:

```
daemon 99    0.0 0.3 1132 500 ?     S Jul11 0:02
/sbin/portmap
tiger  27837 0.0 0.3 1264 492 pts/0 R 17:03 0:00
egrep portmap
```

The first line here informs that portmap is running. If portmapper is not running, you should try to run it manually. The program usually is located in /sbin, but if not, try /usr/sbin. If the program is not available, you have to install it. The name of the package to install may be something like netbase or nkitb.

Having run the portmapper, you can start the needed dæmons: rpc.mountd, rpc.nfsd, rpc.statd, rpc.lockd and rpc.quotad. These dæmons must be started in exactly this sequence. Modern distributions (we have checked Debian 2.2r3, Red Hat 7.1 and SuSE 7.0) have start/stop scripts for the services. If you have trouble finding such scripts, you may take a ready-made one from a SuSE or Red Hat distribution and correct it by hand, or write your own.

In general, rpc.rquotad is needed only if you use the quota system to limit disk space for the users. In recent versions of the kernel, rpc.lockd is called by rpc.nfsd on request but starting it from the script is okay.

After starting all those programs, execute the command **rpcinfo -p**. The output of this command shows what programs are running currently, their versions and other useful information. At least portmapper, mountd and nfsd must be running to enable you to use the NFS server. If you have any problems, you can find help in the Linux NFS-HOWTO document.

### Setting up the NFS Client

First, make sure your Linux kernel supports NFS. The status of your system is reflected by proc directory. To look inside the filesystems file, execute the cat command, like this:

```
tomcat> cat /proc/filesystems
        ext2
nodev   proc
nodev   devpts
        iso9660
nodev   nfs
```

This file shows you what kind of filesystems you may use with your version of the kernel. If the line **nodev nfs** is missing, it may mean your NFS filesystem support module is not installed. In this case, try to execute the command **modpobe nfs**. You must be root to execute it. The output of the command delineates the situation. If module is compiled, the command installs it, and if you repeat the cat command, the clue line will be shown. If your kernel does not support the NFS server, you probably will have to recompile it, but this topic is far beyond the scope of this article.

If you have NFS supported, you can mount files located on the server machine, as we showed in the example. The full format for the mount command is as follows:

```
mount -t nfs server:exp dir local dir options
```

Here *server* is the name of your NFS server, and exp_dir is the full path to the directory to export on the server machine.

Usually this is the end of the story, but to reach maximum productivity for connection via NFS, let's look at options parameters. We will use them later for sure:

- rsize=n and wsize=n: specify the size of the datagram sent or accepted by the NFS client while reading and writing requests, respectively. The default value depends on the version of the kernel and may be set in 1,024 bytes. The larger the piece of food your cat can eat at one time, the quicker it finishes eating. So, the bigger the value you hput ecere, the quicker your work with a remote file will be. If you place 4,096 or even 8,192 here, throughput may be improved greatly.

- timeo=n and retrans=n: answer for reaction on RPC timeouts. If your pet (take a cat, for example) has eaten a lot, it is able to take a new little piece of food, after a short period of time to be digested a little, and eat a lot again after a long period of time. These time intervals are called minor and major, respectively. Although NFS is not a cat, it also has minor and major time intervals, connected with the reaction of the server. Either the Net or the server, or even both, may be down temporarily or inaccessible. If this occurs, the resending starts after timeo tenths of second. Standard value for this minor timeout is seven. So if there is no reply within 0.7 second, NFS client will repeat the same request and double the timeout (1.4 seconds) by default. If there is still no reply, the procedure will be repeated until a maximum, major timeout of 60 seconds is reached. The quantity of minor timeouts that must occur before a major timeout will be reached, is set by the retrans parameter; its default value is three.

- hard and soft: define the reaction if major timeout is reached. In the first case, if default value hard is set, NFS client reports the error “server not responding, still trying” on the console and continues to try to connect. So when the NFS server is back on-line, the program will continue to run from where it was. For the opposite, if value soft is set, the NFS client will report an I/O error to the process accessing a file on an NFS-mounted filesystem, and some files may be corrupted by losing data, so use the soft option with care.

- intr: allows for interrupting an NFS call and is useful for canceling the job if the server doesn't respond after a long time.

The recommended options are intr and hard, and since the latter is the default, intr is enough. With these parameters, we are able to try to increase performance of the NFS connection.

Often we want to do the process of mounting in a hidden way (transparently), perhaps while booting the system. In this case we have to create the

corresponding line in the file named /etc/fstab that the system reads while booting. Any line in that file has as a minimum four fields, usually six. Here's an example:

```
1)device 2)mnt point 3)fs type 4)options
5)dump 6)check order
tiger:/nfs1/home /home nfs rw,hard,intr 0 2
```

The first parameter describes the device to be mounted. In this example, the NFS server tiger exports /nfs1/home filesystem as the home directory. Field number four points to this filesystem as nfs. The last parameters mean the filesystem should not be dumped and checked in the second order. You can find more details in section five of the man pages. It also is a good idea to start rpc.lockd and rpc.statd. These programs usually initialize from scripts. Once you've done those things, you should have the NFS client running.

### Planning to Use NFS

What kind of data can we keep on NFS? The simplest answer is any kind, including the whole root system for diskless machines. In order to plan your system, you have to take into account that all files can be divided into two categories: shareable and unshareable data. The latter must be kept on a particular machine. For instance, device (and lock) files are not shareable. The size of some directories is stable for some and unstable for others.

Assuming we have a small system, we show /home as typical shareable data that should be located on the server host. On medium and large systems, it is useful to subdivide user home directories by using such names as /home/students, etc. The size of the $HOME directory is not static, of course, but to limit disk space for many people, you may set up a quota system on the servers.

What about another directories? Some of them have a static size, some change permanently. Shareable static directories include /usr and /opt, while unshareable static directories include /etc and /boot. Static variable directories include /var/mail and /var/spool/news, whereas unshareable variable directories include /var/run and /var/lock.

Many people do not put /usr/bin and /bin files on NFS because of speed, but this is not very critical. We believe an ordinary user does not notice any difference in calling a program from an NFS directory vs. from a local one. Therefore, if disk space is critical, you may mount mostly everything via NFS, importing files from the server machine while keeping space on client machines for other files, such as MP3s or two OSes.

Although you may keep everything on the server, the standard solution is to keep only the $HOME directories and documentations there, including man pages and other documentation. Sharing by means of NFS software that is used rarely and located typically under /opt and /usr/local, is also a very good idea. Two subdirectories should be shared via NFS: /usr/doc and /usr/share.

Now, how do we use the mount parameters in these cases? As discussed previously, the recommended options for $HOME are intr and hard. In this case, you can be sure the data will not be lost. For /usr/doc and similar subdirectories, however, this may not be the optimal choice. If you access man pages while the Net is overloaded, or even if the NFS server is down, your machine will wait until it is able to re-establish contact with your NFS server. In this case, using the soft option is better; the disconnect message will appear on the screen and the operation will be canceled. This option may be useful for noncritical data of any kind, including news and FTP directories. However, the standard choice is not to access /var/mail and /var/news via NFS; use special protocols such as imap, pop and nntp.

### Improving the NFS Connection

If you are concerned about the speed of the NFS connection, you also can play with the rsize and wsize options. The default size is set by the kernel and may or may not fit. For instance, either your Ethernet card or the kernel is not able to handle large blocks, something that usually happens with old hardware and old kernels. If you have a newer card, a larger size might help you reach better performance. In some cases you can increase the speed by five or more times, but you have to do some experiments with your network.

The easiest way of calculating the speed is to create a file located on NFS volume and find the time of creating. This is just an experimental file that may hold any kind of data, including zero sequence, so we will create it by means of the dd command, using /dev/zero as the source file. Command time calculates time of creating.

How large should the temporary file be? The rule is to have two or three times the size of the memory on the server machine. If you have 2MB of RAM there, the file should be 4MB. Sometimes it is hard to find a place for a such file, but in many cases 256MB on your NFS disk machine should be fine. To calculate the free area on your disk, use the command df. Then, assuming you are working on a client machine, unmount it and mount again, put the magic value 1,024 for rsize and wsize parameters. Then execute the dd command with time:

```
time dd if=/dev/zero of=/home/tempo bs=16k count=16384
```

We use /dev/zero as the source for /home/tempo in order to avoid calculating the time of reading the source. This command sends 16,384 16KB blocks of null bytes, creating a file with a total size of 16,000 bytes × 16,384/1024 kilobytes = 256 kilobytes or 256MB. Write the time down. Then calculate the time of reading, sending the file to /dev/null:

```
time dd if=/home/tempo of=/dev/null bs=16k
```

Record the time of reading, and then delete the temporary file:

```
rm /home/tempo verbatim
```

Repeat the describing procedure at least three times and calculate the average reading/writing time. Unmount your client host:

```
umount /home verbatim
```

Make sure you have NFS unmounted; it complains if there are errors and keeps silent if not. Then mount it again, using the new value 2,048 for rsize and wsize, and repeat the procedure above, again and again, up to 32,768.

Observe the result. The faster the transferring, the shorter the time. Find your optimal value and put it into /ets/fstab. Keep the paper with the calculations in a safe place.

There are also a few tips to help you make this list shorter. If you have a new network card and are sure your NFS server runs version 3, you may start from the value 32,768 and go down. This value should not be tested with NFS server version 2, however, since it only works with older kernels and some Solaris and SunOS machines.

We found the best result for kernel 2.2.19 to be value 4,096, so you may start from it and try the nearest, 2,048 and 8,192. For the new kernel, this value (in our case) was 8,192, but very often kernel 2.2.19 worked better in comparison with 2.4.6. It strongly depends on your hardware, so be very careful if you want to switch your NFS server into new kernel release.

The optimal value for client machines that use NFS over TCP/IP is 1,024 or less. This value also works if you have a router between a client and the NFS server.

### A Word about NFS Versions

While the mount command for the NFS filesystem is working, the version of the protocol is being recognized and a warning message may appear on the screen. Usually it says the NFS version of the server is older than the kernel mechanism.

The idea of sharing files through the Net is as old as the Net itself, so there are concurrent solutions lists. To improve NFS try version 4, or NFSv4 for short, which was introduced in the year 2000. The inventor of NFS, Sun Microsystems, is a sponsor for the Linux NFSv4 Project. It has, of course, all the features of previous releases but supports security by extending the basic RPC security mechanism, so the weakest point of the previous version has been solved. The other weak point, lack of internationalization, does not exist anymore. Speed of transferring has improved due to caching. We also can avoid those procedures of calculating the optimal parameters because the new mechanisms of improving access are supported. Putting this version into practice means going without dust and noise—nowadays the Linux NFSv3 server supports file locking —an NFSv4 feature.

So, don't forget about the old vine on the Net, NFS. It is still not the vinegar of history.

Resources

**Olexiy Tykhomyrov** (tiger@ff.dsu.dp.ua) has been using Linux since 1994. He works for the Department of Experimental Physics at Dnepropetrovsk National University and teaches physics and communications. He loves his son Misha who calls him Tiger because some of his father's students are afraid of him. Tiger likes swimming and traveling.

**Denis Tonkonog**, a former student of Tiger, works in the same place and also likes traveling and fishing with a gun. Friends call him Black Cat but nobody knows why. He may be reached by e-mail at denis@ff.dsu.dp.ua

Archive Index Issue Table of Contents

Advanced search

# Improving Server Performance

**Chen Chen**

**David Griego**

Issue #93, January 2002

How to improve your server's performance by offloading a TCP/IP stack from a Linux-based server onto an iNIC.

Looking to improve the performance on your high-end Linux server? Is your Linux system connected to a high-speed network? Are your servers spending too much of their resources on processing the TCP/IP stack and Ethernet frames? These are just some of the problems in today's network environment. These problems arise because TCP/IP traffic on the Internet and on private enterprise networks has been growing dramatically for the past decade, and there is no sign that growth will be slowing down any time soon. The widespread global adoption of the Internet and the development of new networking storage technologies such as iSCSI are driving networking speeds even faster. Although the processors being manufactured today are gaining speed at an astonishing pace, it is likely that network-related growth will continue to outpace the increasing processor speeds, slowing servers from their primary tasks to process network packets.

Intel has developed a prototype that offloads an entire TCP/IP stack from a Linux-based operating system onto an intelligent network interface card (iNIC).

## The iNIC Design

The iNIC contains a real-time operating system (RTOS) and an entire TCP/IP version 4 stack. An I/O processor (IOP) on the iNIC processes all of the network packets allowing the host processor to process other tasks. To accomplish this division of labor, a thin layer of logic is needed on the host side to route all the network traffic through the iNIC.

## Socket Offload

This technology is based on the intelligent I/O (I2O) architecture, which is already incorporated into the Linux 2.4.x kernel. Figure 1 is an I2O primer for those who are not familiar with I2O. The I2O specification is a message-based communication mechanism between a host operating system (OS) and I/O devices that are sitting on an IOP. The IOP runs an intelligent RTOS (IRTOS), which contains device-driver modules (DDMs) for each connected device. To obtain portability, I2O uses a split-driver model. The specification defines a set of abstract messages for each supported device class (i.e., LAN, tape, disk). The host OS uses the abstracted message layer to communicate with DDMs running on an IOP. The DDM translates these I2O messages to hardware-specific commands. To communicate with an I2O device, the host OS must have a driver that knows how to translate OS device commands to I2O device class commands. This module in the host OS is referred to as operating system module (OSM).

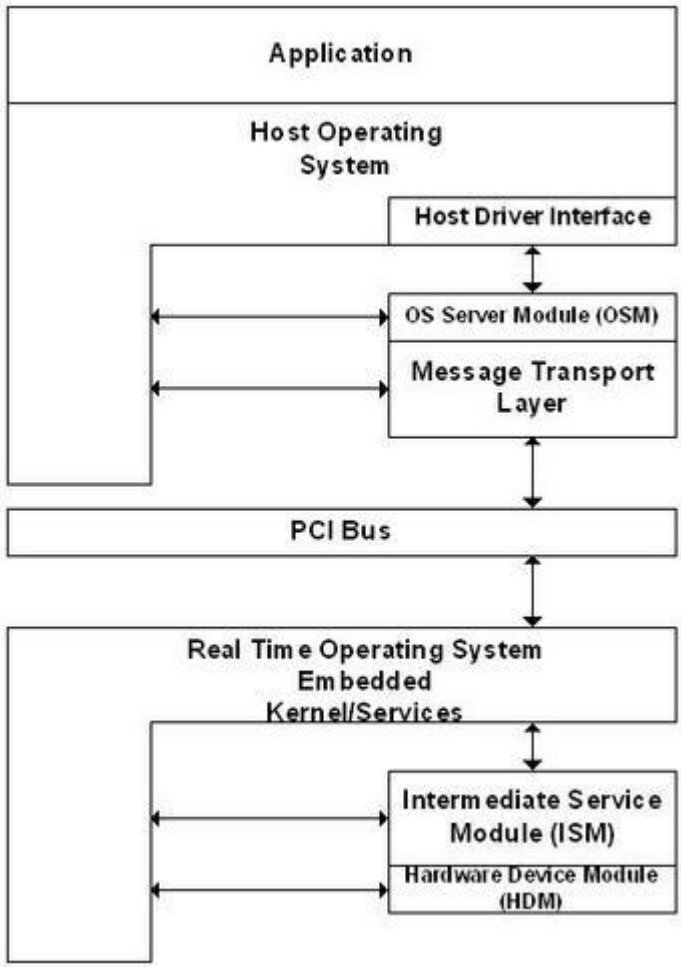



Figure 1. I2O Architecture

The solution is based on an extension of this architecture with the creation of the socket class. Changes were made to the I2O architecture to increase

performance and minimize latencies usually associated with a split-driver model.

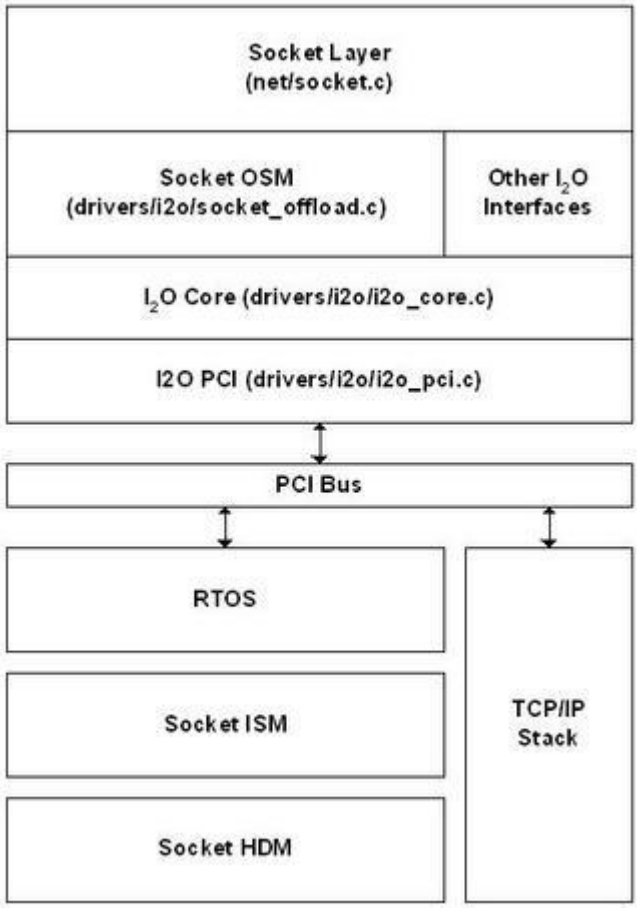



Figure 2. Socket Offload Architecture

The socket class defines messages needed for communication between the host OS and the DDM. The two drivers, OSM and DDM, communicate over a two-layer communication system. The message layer sets up the communication session, and the transport layer defines how information will be shared. The DDM is composed of two modules: intermediate service module (ISM) and hardware device module (HDM). The ISM provides the full functionality of the TCP/IP version 4 stack. The HDM is the device driver for the iNIC. The socket OSM is unlike any other network device in Linux. Normal network card drivers are protocol-independent and interface with the Linux kernel at the network application program interface (API). The socket OSM, on the other hand, will interface directly below the socket API. This allows the necessary socket services to be offloaded onto the IOP running the socket class ISM. The socket OSM replaces the services that the TCP/IP stack provided to the kernel, thereby providing necessary interfaces to the Linux kernel. It also transmits and converts socket requests and data in the socket offload format to the iNIC running the TCP/IP offload stack.

## Socket OSM

The OSM is divided into the following subsystems: user interface, message interface, kernel interface and memory management.

The user interface replaces the af_inet socket layer in the kernel. It provides feedback to the users' programs exactly as the native (non-TCP/IP offloaded) kernel would provide.

The message interface provides the initialization and control of the socket offload system. It translates the user socket requests to the socket messages.

The kernel interface provides kernel services to the OSM. This is the point at which the OSM provides any services normally provided to the kernel by the TCP/IP stack. This subsystem was designed to minimize the modifications needed for the Linux kernel.

The memory management module provides the buffer pools needed for data transfer to and from the user-space applications. Memory management was designed to 1) minimize the number of data copies and DMA requests, 2) minimize the host interrupts, 3) avoid requiring costly physical-virtual address mappings and 4) avoid overhead of dynamic-memory allocation at runtime. Two pools of DMA-capable data buffers are maintained in the OSM. The transmit buffer contains the data headed for the iNIC; the receive buffer receives the data into the kernel from the socket device.

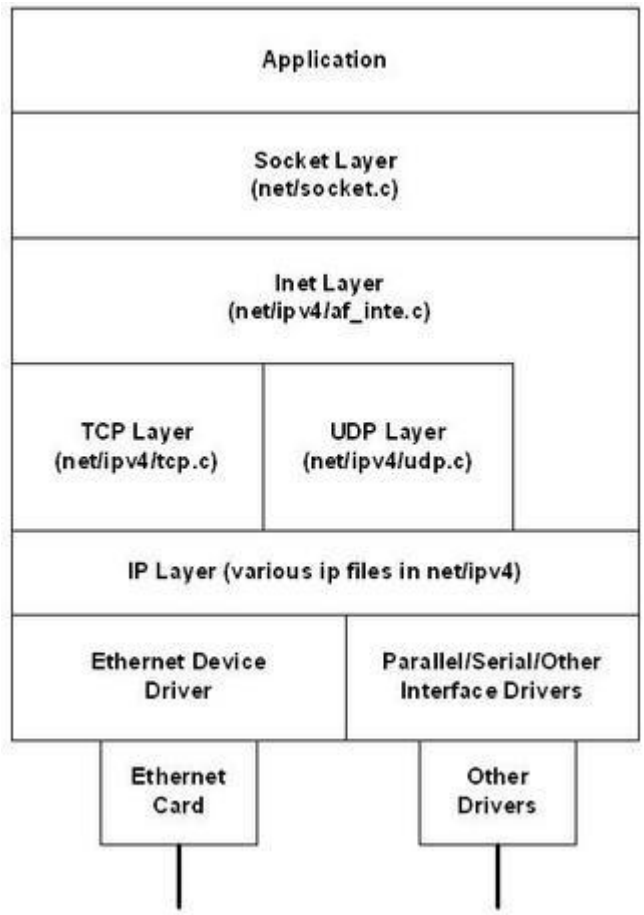



Figure 3. Linux TCP/IP Stack

As shown in Figure 3, Linux network components consist of a layered structure. User-space programmers access network services via sockets, using the functions provided by the Linux socket layer. The socket structure defined in include/linux/net.h forms the basis for the implementation of the socket interface. Below the user layer is the INET socket layer. It manages the communication end points for the IP-based protocols, such as TCP and UDP. This layer is represented by the data structure sock defined in include/net/sock.h. The layer underneath the INET socket layer is determined by the type of socket and may be the UDP or TCP layer or the IP layer directly. Below the IP layer are the network devices, which receive the final packets from the IP layer.

The socket OSM replaces the INET socket layer. All socket-related requests passed from the socket layer are converted into I2O messages, which are passed to the ISM on the IOP.

### Embedded Target

The embedded system software consists of the messaging layer, TCP/IP stack, device driver and RTOS.

The messaging layer is the portion of the software that takes messages from the OSM, parses them and makes the socket call into the TCP/IP stack. This

layer also takes replies from the network stack and sends the appropriate reply to the OSM. To improve performance and minimize the effects of latency inherent in split-driver systems, the messaging layer batches, replies and pipelines requests.

The embedded TCP/IP stack is a zero copy implementation of the BSD 4.2 stack. It provides all of the functionality of a networking stack to the messaging layer. Like all the software that runs on the IOP, the stack has been optimized for running on the Intel 80310 I/O processor chipset with Intel XScale microarchitecture (ARM-architecture compliant). Benchmarks were performed on the TCP/IP stack during optimization to ensure that it would perform well across all sizes of data traffic.

The HDM was written to take advantage of all the offloading capabilities of the NIC hardware. This includes TCP and IP checksums on transmit and receive, segmentation of large TCP packets (larger than 1,500 bytes) and interrupt batching supported by the chip. The NIC silicon chips supported were the Intel 82550 Fast Ethernet Mu.pngunction PCI/CardBus Controller and the Intel 82543GC Gigabit Ethernet Controller.

The RTOS is a proprietary OS that has been designed for the demands of complex I/O operations. This OS is fully I2O-compliant. It was chosen in part because of the willingness of the designers to make modifications to the OS for the prototyping efforts.

As described before, the socket calls made by the application layer are converted into messages that are sent across the PCI bus and to the I/O processor. This embedded system is a complete computer for performing I/O transactions. It consists of a processor, memory, RTOS and a PCI bus. Because it is designed for I/O, it will minimize the effects of context switching. Once a message reaches the IOP, it is parsed. The socket call that was requested by the application is then called on the embedded network stack. A reply message is sent to the OSM once the socket operation is completed.

## Benchmark Results

The benchmark tests that were run using the prototype showed that the offloading of the TCP/IP stack significantly reduced both CPU utilization and the number of interrupts to the host processor. With a heavily loaded machine, the offloaded stack was able to maintain overall network performance and host CPU cycles were able to remain dedicated to the workload applications. In a native machine, the host processors were interrupted far more frequently, and the network application suffered from CPU resource starvation resulting in the network performance degradation.

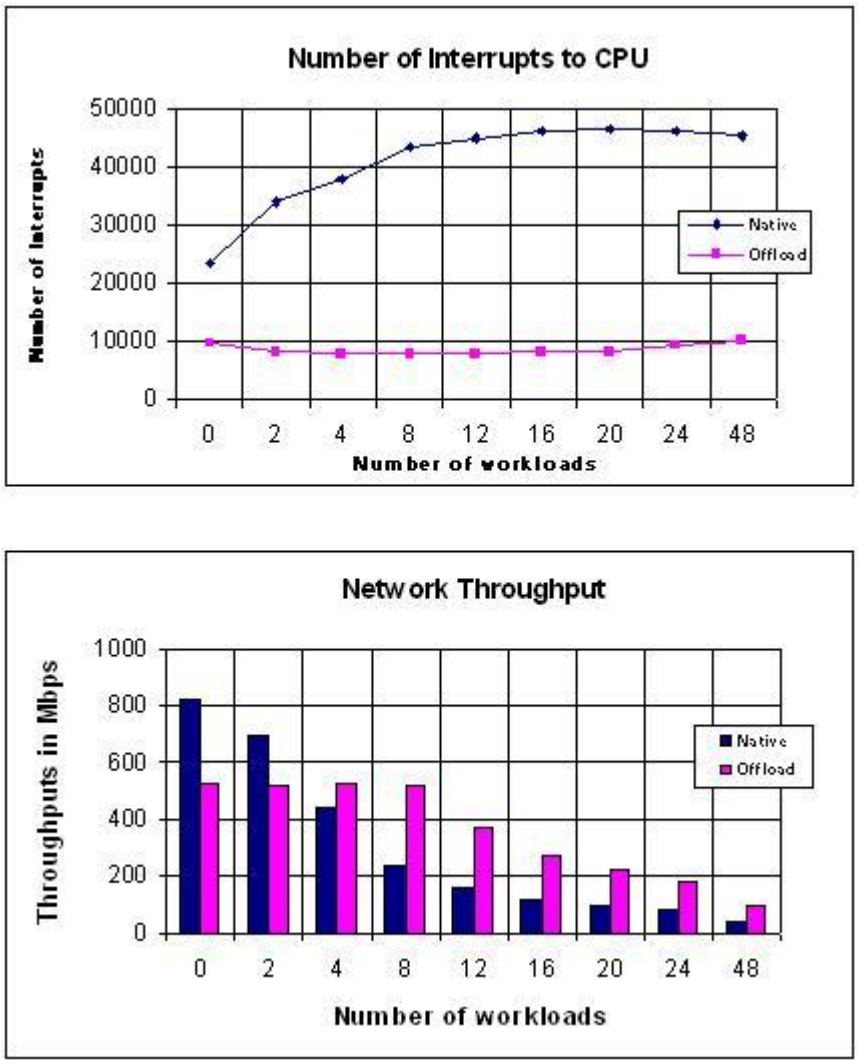



Figure 4. Benchmark Results

## Future Direction

As the subject of iSCSI (storage over IP by encapsulating SCSI in a TCP/IP packet) starts to heat up, desire for minimizing network overhead will continue to grow. Efforts used in moving the TCP/IP stack to an IOP quickly could shift to providing a full-featured TCP/IP stack at the back end of an intelligent iSCSI adaptor. This would minimize the impact of iSCSI to a Linux platform by making it available via the normal SCSI API. To compete with Fibre Channel, iSCSI must provide comparable performance.

Another future enhancement is that embedded Linux will be used for the RTOS. At the start of this prototyping effort, an Intel i960 RM/RN processor was used, and embedded Linux was not available. Since then, the Intel XScale microarchitecture has been introduced, enabling the adoption of the embedded Linux that is available for Intel StrongARM core. Porting of Linux-based StrongARM Linux to the XScale microarchitecture will be completed by the end of the year.

There were several goals behind this prototype effort: 1) to demonstrate that the enhanced performance achieved by offloading network tasks from the host processor reduces the host processor cycles otherwise consumed by processing of network data, 2) to show that the use of specialized software on the iNIC performs the same networking tasks while maintaining overall network performance and 3) to enable the use of I/O processors to work in conjunction with the host processors to handle the network traffic, thereby maximizing performance of a Linux-based server at minimal cost.

Offloading the TCP/IP protocol to a specialized networking software environment using embedded processors is an effective way of improving system performance. With the advancement of high-speed network deployments and adoption of network storage, TCP/IP will inevitably play an important role.


## Acknowledgements

Technical contributors: Dave Jiang, Dan Thompson, Jeff Curry, Sharon Baartmans, Don Harbin and Scott Goble.



**Chen Chen** is doing research on advanced I/O applications at Intel. He can be reached at chen.chen@intel.com.



**David Griego** has been a Linux enthusiast for more than three years. He works at Intel in Advance Development Engineering. He can be reached at david.a.griego@intel.com.



Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL



# Entity Beans

**Reuven M. Lerner**

Issue #93, January 2002

Entity beings? No, beans. Learn to write 'em, connect 'em to a database and access 'em via their cousin, the session bean.

Last month, we began to look at Enterprise JavaBeans (EJB), the centerpiece of Sun's J2EE (Java 2 Enterprise Edition) standard for server-side web applications. While neither the Java language nor the J2EE specification are open standards, increasing numbers of Linux advocates have begun to use them to write server-side web applications. The fact that J2EE appears to be the only mainstream alternative to Microsoft's .NET framework makes it even more appealing to many.

Enterprise JavaBeans come in two basic flavors, known as session beans and entity beans. Session beans typically model processes and lack any state, allowing us to place our business logic in EJB, rather than in our servlets or JavaServer Pages (JSPs). The calculator object we designed last month, which allowed us to multiply two numbers, was a simple example of a session bean with a single method.

Entity beans are meant to contain state, possibly even complex state. This state normally reflects the contents of a row in a relational database, with the bean managing its own object-relational mapping (bean-managed persistence or BMP) or allowing EJB to handle this task instead (container-managed persistence or CMP). The EJB container also provides transactions, giving us all-or-nothing operations in our objects as well as the database.

This month, we write a simple entity bean, connect it to a database and access it via a session bean from a Java application. We will use the open-source JBoss EJB container (released under the GNU Lesser General Public License, aka LGPL), but the code should work with little modification on any J2EE server that supports EJB.

## Creating Entity Beans

As we saw last month, writing a session bean really means writing three Java classes:

- The bean class performs the actual work.
- The remote interface has methods that match those on the bean class and is our proxy to the bean.
- The home interface helps us to create new instances of the bean class, as well as search for beans matching particular criteria.

We need to implement all three of these classes for an entity bean. In addition, we often need to implement a fourth “primary key” class. While this month's example does not need to define a primary key class, we will do so in the interest of completeness.

Most EJB applications will end up using at least one entity bean (to model the data) and at least one session bean (to implement the business logic). Given that a core idea of object-oriented programming is to put data and code in a single package, it seems a bit strange to split entity and session beans in this way. But this strategy does seem to work overall and makes it relatively easy to split work among multiple people, once the specification has been agreed upon.

## Working with JAWS

J2EE is a specification; the actual implementation of that specification depends on whoever has written the server. One of the most important parts of a J2EE application server is the object-relational mapper, which transparently turns Java classes into rows of a relational database table (and vice versa). An object-relational mapper should remain as invisible as possible, allowing us to change our back-end storage from Oracle to MySQL without modifying our Java code.

The JBoss object-relational mapping system is known as JAWS and normally requires very little configuration. However, it can be instructive to look at the JAWS configuration file (standardjaws.xml, in the JBoss conf/default directory) to see what's happening behind the scenes.

The definitions at the top of standardjaws.xml set parameters for the entire JBoss server. In this way, we indicate which database we want to use; the HyperSonic database is included with JBoss, and we will use it for this month's examples.

The core of standardjaws.xml is the multiple <type-mapping> (singular) sections, which connect each <java-type> to a <jdbc-type> and an <sql-type> for each database. Since our EJBs do not create tables or write SQL explicitly, it's important that these values be accurate. You may be able to increase the efficiency or flexibility of your application by modifying these values. However, remember that modifying JAWS after you already have inserted data into a database may lead to confusion, corruption or errors.

If you simply want to get started with EJB, then you won't need to modify standardjaws.xml at all. Rather, you'll need to modify jboss.jcml, an XML file that defines the different managed beans (MBeans) that JBoss uses for system configuration and control.

The file jboss.jcml includes support for HyperSonic and InstantDB; in order for it to work with HyperSonic, I had to remove any reference to InstantDB from jboss.jcml. I did this by editing the “JDBC” section of jboss.jcml, removing the mention of org.enhydra.instantdb.jdbc.idbDriver from the “Drivers” attribute for the JdbcProvider MBean and the entire XADataSourceLoader <mbean>, whose service is XADataSource and whose service name is InstantDB.

Once you have removed all mentions of InstantDB from jboss.jcml, start up JBoss:

```
cd $JBOSS_DIST/bin
sh run.sh
```

## Writing Our Entity Bean

Our entity bean models a single book, where each book has a title, an author, a publisher and a price. For the sake of simplicity and space, we will ignore the possibility that a book might have multiple authors or publishers. We also will avoid normalizing the data, which would mean having instance variables that are themselves entity beans.

We begin by implementing the BookBean class, depicted in Listing 1 [available at ftp.linuxjournal.com/pub/lj/listings/issue93/5577.tgz]. BookBean is a typical simple bean class definition for a container-managed entity bean; it defines a field for each column in the database that we want to trace, including an integer “id” field that serves as a primary key.

We must define the ejbCreate() method to match the signature of the create() method on the home interface. Each time someone invokes create() on the home interface, the EJB container invokes ejbCreate() on our bean class with the same arguments. **ejbCreate()** is where the real creation action is; while a CMP entity bean doesn't need to worry about handling the object-relational mapping, it does need to set its instance variables to appropriate values.

Other than ejbCreate(), the only methods we must write are the “getter” and “setter” methods for each field, such that other objects can retrieve or modify the field's value. Each method is pretty simple in our example, returning or modifying the value of an instance variable.

Our remote interface, shown in Listing 2, is called Book.java, and its API is almost identical to the bean class. Applications normally will talk to the remote interface; if something goes wrong, it throws a RemoteException.

Listing 2. Source code for our remote interface, Book.java—this class mirrors the public methods defined by our BookBean class.

We also define a home interface, shown in Listing 3, with a create() method that creates a new instance of Book (and implicitly, a new row in our database table) when handed all of a book's details. If we were so inclined, we could offer users multiple versions of create(), each of which would take a different number of arguments.

Listing 3. Defining a Home Interface

Notice how our create() method requires that we provide an explicit primary key. Experienced database programmers know that primary keys should be hidden from view, and most databases have a way to automate this; PostgreSQL's SERIAL type, MySQL's AUTO INCREMENT and Oracle's sequences are common solutions to this problem. Unfortunately, there is no easy way to use such automatically generated primary keys within EJB. Therefore, we must set it explicitly (as in this month's examples) or use an external value, such as the ISBN, which could be a String. This is one of the most surprising things that I've found about EJB, and I hope that future versions of the specification will remedy the situation.

The findBy...() methods allow us to locate and retrieve instances of Book. Invoking findByPrimaryKey(5) returns an instance of Book with the primary key 5. All of the findBy...() methods are implemented by the EJB container, saving us from having to do so ourselves. The findAll() method returns a collection of all objects of this type (i.e., all rows in the database table), allowing you to iterate through them.

Unfortunately, automatically defined findAll...() methods use simple equality checks. We cannot use regular expressions or other techniques to search for books whose author field begins with O, or whose publisher begins with A and ends with M. Instead, we must use findAll(), iterating through the returned collection and filtering through them as necessary.

Finally, our primary key class (BookPK.java), shown in Listing 4, defines a single instance variable (id) that acts as our primary key. The equals() method indicates whether two instances of BookPK are identical, allowing the system to compare two instances of Book. The hashCode() method must return a unique value for each instance, which can be the id in this particular case. The toString() method must return a string version of the primary key, which simply returns String.valueOf(id) in our class.

Listing 4. BookPK.java, the Primary Key Class for Our Entity Bean

Because all four of these classes are in the il.co.lerner.book package, I placed all four source files (Book.java, BookBean.java, BookHome.java and BookPK.java) in the directory $BOOK/il/co/lerner/book, where BOOK is the root directory of this project.

### Deployment Descriptor

Now that we have defined our entity bean, we need to describe it to the EJB container. Our deployment descriptor, a file named ejb-jar.xml, is shown in Listing 5. We place ejb-jar.xml in $BOOK/il/co/lerner/book/, alongside the Java classes that will form our entity bean; when we build the bean with Ant, it will be placed in a subdirectory named META-INF.

Listing 5. ejb-jar.xml, the Deployment Descriptor for Our Entity Bean

The most interesting parts of ejb-jar.xml for an entity bean are the <persistence-type> section (set to “Container” for CMP), the <prim-key-field> and <prim-key-class> sections (in which we name the class of our primary key), and the <cmp-field> sections (which describe the container-managed fields to JBoss).

The deployment descriptor is a standard part of EJB and should work across all EJB servers and containers. However, it does not address all of the runtime configuration issues. For JBoss to work correctly, we thus include a file named jboss.xml that tells the server where we can find the beans on the network. A copy of jboss.xml, which we place alongside ejb-jar.xml in $BOOK/il/co/lerner/book/, is shown here:

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss>
  <enterprise-beans>
    <entity>
      <ejb-name>Book</ejb-name>
      <jndi-name>Book</jndi-name>
    </entity>
  </enterprise-beans>
</jboss>
```

## Application

The source code for our simple test application, UseBook.java, is presented in Listing 6 [available at ftp.linuxjournal.com/pub/lj/listings/issue93/5577.tgz] and demonstrates how much you can do with very little code. It defines only a main() method and immediately goes about working with the EJBs. First it grabs a JNDI context and looks up the Book bean that we have defined. That allows it to create an instance of BookHome, which in turn lets us create a new instance of our Book bean:

```
Book book = home.create(testPrimaryKey, "Book title",
                        "AuthorFirst AuthorLast",
                        "PublisherName", 10.50);
```

As you can see, we've hard-coded the values we will add to the database, including the primary key. This is obviously unacceptable in the real world; a nonexample application would have taken the primary key (and other values) from a file, the command line, an environment variable, an HTML form on the Web or would have generated them automatically.

Notice how we never have to create SQL tables, insert rows or retrieve them. Our back-end persistent storage is presumably a relational database, but our Java client neither knows nor cares.

Once it has inserted a new row into the table, UseBook modifies some of its values (by setting instance variable values) and then retrieves all of the instances of Book in the database (with findAll()). Along the way, it sends status messages to System.out, which are printed on the console.

## Ant Configuration File

We use Ant, the Java replacement for the standard make program, to compile and install our program. Listing 7 [available at ftp.linuxjournal.com/pub/lj/listings/issue93/5577.tgz] shows our build.xml file, which compiles each of the .java source files from $BOOK/il/co/lerner/book, turns them into a jar file along with the deployment descriptor and JBoss runtime configuration file, and then installs the files into the JBOSS directory. If Ant is installed in ANT, you can compile all of the files, install them into the JBoss “deploy” directory and invoke UseBook.main() with:

```
$ANT/bin/ant use-book-ejb
```

As soon as JBoss notices the new (or updated) jar file, you will see output on the screen indicating what the server is doing. The output from Ant, by contrast, will display all of the items sent to System.out from within main(), including the status messages in Listing 6 that indicate what we have done. Each time you

compile and run main() with a new ID value, a new row will be inserted into the database table.

### Is EJB the Future?

Sun would like you to believe that EJB is the future of all server-side web applications. Microsoft, of course, is doing its utmost to convince you that .NET is the real future. How do independent developers fare in this war of giants, and what should free software advocates do?

The good news is that J2EE is an excellent architecture and philosophy. It's neither the most elegant nor the most flexible, and it does lock you into a single language. But overall, I'm impressed with J2EE and see it as an important milestone in the world of web application programming.

Unfortunately, there are a number of issues with J2EE that arise every time I use it to write an application. The first issue is that neither Java nor J2EE are open source, despite the fact that Java is free of charge and JBoss is licensed under the LGPL. Sun generally has been an honest player, but they are a commercial company with their own interests. Open-source advocates should not be surprised if and when Sun restricts the use of code or specifications.

In addition, the departure of Enhydra Enterprise appears to leave JBoss as the only open-source J2EE application server on the market. JBoss isn't officially certified as J2EE-compliant, however, because the JBoss team cannot pay for official certification. This strikes me as rather shortsighted of Sun; perhaps they could offer free or inexpensive certification for servers released under the GPL or LGPL, which (unlike the Berkeley license) ensure that no company could ever turn a certified server into a proprietary product. Official J2EE certification isn't important to me, but there are many CEOs and CTOs who do require it, which means that JBoss is often ignored for no good reason.

Java and EJB are complex, and it will take time for a programmer to learn about them. But the complexity of Java and EJB programming is dwarfed, in my experience, by a confusing array of configuration files, new terms, environment variables and other items that are unique to Java. Better documentation would certainly help, but it seems to me that a Java analogue to CPAN, which would make server-side Java configuration easier, would let programmers concentrate on programming rather than system maintenance issues.

Finally, the one part of .NET that really appeals to me is its relative openness to different programming languages. By definition, J2EE requires Java, which is often the right choice but should never be the only choice. SOAP and XML-RPC make it possible to bridge the gap between languages but without the nice transactioning and object-relational mapping that EJB brings to the table. For

now, it seems that the only way for Python to speak to EJB is via SOAP or XML-RPC (or Jython), but I would love to see other possibilities in the future.

## Conclusion

EJB is an impressive technology, doing far more than the simple object-relational mappers Alzabo and DODS. From my experience, working with EJB is more of a managerial and logistical headache than a technical one. Learning EJB is a good idea for all web application developers; it's clear that this standard is making serious inroads in the industry, and many serious applications will be built using EJB in the future. Having certified open-source implementations will make it even easier for programmers to try out EJB, and I encourage Sun to move in this direction as soon as possible.

Next month, we'll switch gears to begin looking at Zope, a very different type of web application framework written mostly in Python. Zope has become quite popular in the last few years and is often seen as the killer app that will bring Python to the forefront of programming languages. We'll take a look at Zope and start to examine how we can use it to write our own applications.

Resources

**Reuven M. Lerner** owns a small consulting firm specializing in web and internet technologies. He lives with his wife Shira and daughter Atara Margalit in Modi'in, Israel. You can reach him at reuven@lerner.co.il or on the ATF home page, www.lerner.co.il/atf.

Archive Index Issue Table of Contents

Advanced search

# Networking for Pleasure

**Marcel Gagné**

Issue #93, January 2002

Challenge your old friends, wherever they live, to a game of Risk or Monopoly, across the network, across the world.

The wonderful thing about the Linux community, François, is that there really is a community. In some cases that community goes back to before Linux was born. Today, more than ten years after Monsieur Torvalds released Linux to the world, the tradition continues, those common bonds among computer enthusiasts only heightened by the Internet's *toujours là* nature. Networks, François—that is what makes this community strong.

*Mais oui*, François, you are correct. While this whole notion of networks leads itself to business communication, I think you will agree with me when I say the real keyword is communication. That means bringing people together, and what better way to bring people together than through games. We are a playful species, *mon ami*. I believe human evolution has brought us to this point, and the real purpose of all this networking technology, François, is to make it possible for people to play games with others around the world or across the block—that and to exchange wine-tasting notes via e-mail of course.

*Ah, bonjour mes amis*! François! To the wine cellar. *Vite*! Bring up the 1997 Mendoza from Argentina for our guests. Please, sit down, *mes amis*. Be comfortable. François and I were just discussing ways to unwind with friends who are far away. You know, when I was younger, we would sit around the table with a glass of wine, chatting loudly and enjoying a classic board game. With today's networking technology and your Linux systems, we can still do that, even when our friends are on the other side of the planet. Ah, François. *Merci*. Please, pour for our guests.

The first such game I want to look at is an old favorite of mine. You know the one—five ships are assembled on a grid representing a strange ocean. Two players face off, firing shots blindly:

“E-7.”

“Miss.”

“E-8.”

“Aw, you sunk my destroyer!”

This game even has been played on scraps of paper with the grids and ship positions drawn in. Now, if your long-lost friend is at the end of an internet connection on some far shore, you still can experience the joys of the game. In fact, it is even better. Ricardo Quesada is the author of *Batalla Naval*, a network-enabled game that takes this classic into a new realm. With this version of the game the two-player limit is gone, meaning that more can join in on the action. Now you have more to worry about than a single opponent. The excitement begins by visiting the *Batalla Naval* web site at batnav.sourceforge.net and downloading the latest source. Building the software will be familiar to many of you:

```
tar -xzvf gbatnav-1.0.2.tar.gz
cd gbatnav-1.0.2
./configure
make
make install
```

*Voulez-vous jouer*? In order to start a game, you must first execute the server program, gbnserver. It is also possible to connect to an existing server running somewhere else on the Internet. On the machine that launches the server, a window will pop up showing client connection statuses ranging from 0 through 9. At this point, you can join the game using the gbnclient program. The first thing you will see is a connection box, asking for server and port number. If you started the server by simply typing the command name, that port will be 1995. You also can choose a name for your player at this time. The default is your login name.

When you are ready, place your ships on the grid (a total of ten ships in this incarnation of the game) and click Send ships when you are happy. You'll need at least two players, but you also can fire up a robot to play against. When you are ready, and you have all the players you want, click on the Start game button.

So now we play over the network, whether it be across the room or around the world. But what about the friendly, energetic banter? What makes this feel like high-tech-meets-old-fashioned play is (strangely) something that takes advantage of the networked nature of the game—a chat interface. During the course of the game, you can let friendly banter flow back and forth among the assembled players by typing your messages in the text box at the bottom of the client. All messages will be prefixed with the name of the person who sent the message. Figure 1 shows a game of *Batalla Naval* in progress.

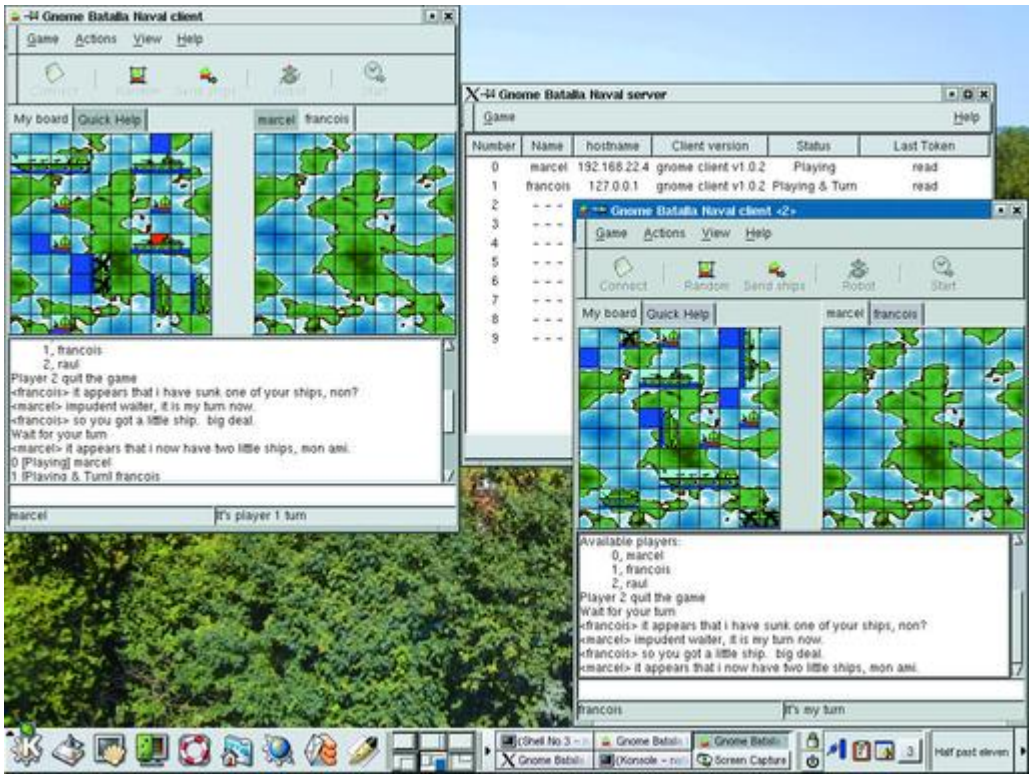


Figure 1. A Fierce Game of

The next item I want to cover appeals to the capitalistic land baron in all of us. It is one of the most popular games of all time, the classic real-estate trading and selling game called *Monopoly*. Over the years, it has spawned many variations including boards that represent a number of major cities throughout the world. Atlantic City is but one locale for the virtual real-estate mogul.

You'll find a couple of variations of this classic game out there, and we will get to that shortly. It is possible to connect to existing servers on the Internet, but there were no permanent servers when I checked into this, so the best bet is to run your own server (particularly if you want to play with your own crowd). The program that allows you to run such a server is Rob Kaper's monopd. The address for downloading the monopd dæmon program is sourceforge.net/projects/monopd. You will, however, need one other piece before you actually can assemble the game dæmon. This piece is the libcapsi_network library (once again from Rob Kaper), and it is also available from SourceForge, but at the

following address: sourceforge.net/projects/libcapsinetwork. Let us start there. Begin by extracting the libcapsi source:

```
tar -xzvf libcapsinetwork-0.0.13.tar.gz
cd libcapsinetwork-0.0.13
./configure
make
make install
```

Now that we have the prerequisite library, we can follow essentially the same steps and create our monopd server dæmon:

```
tar -xzvf monopd-0.2.0.tar.gz
cd monopd-0.2.0
./configure
make
```

To start the dæmon, type **monopd**. The program listens on port 1234, and as it works you'll see little messages like **entering monopigator**. You conceivably can run the monopd program at this time, and it will run happily in the background, but we do need something else before we can start having real fun. *Oui*, *mes amis*, I agree that dæmons are fun, but they do not make a board game, *non*? What we need are clients. The client we have on today's menu is Eric Bourget's gtkmonop. You can get gtkmonop from gtkmonop.sourceforge.net:

```
tar -xzvf gtkmonop-0.2.0.tar.gz
cd gtkmonop-0.2.0
./configure
make
make install
```

When the client starts up, you'll notice three tabs. One is labeled Game Board, and it displays the game board. Amazing, is it not? The second is Assets, and from it you may wonder upon the wealth you have accumulated in the form of properties. The third is where you begin. Clicking this tab will provide you with a dialog for choosing a server (if you started it on your own machine, this is likely “localhost”), pick a nickname for the game and click Connect. When a second player joins the game, you are ready to begin.

As with the previous item on tonight's menu, *Batalla Naval*, *gtkmonop* provides a chat window so that you may (ahem) chat with your game partner or partners. Have a look at Figure 2 for a screenshot of a *gtkmonop* game in action.

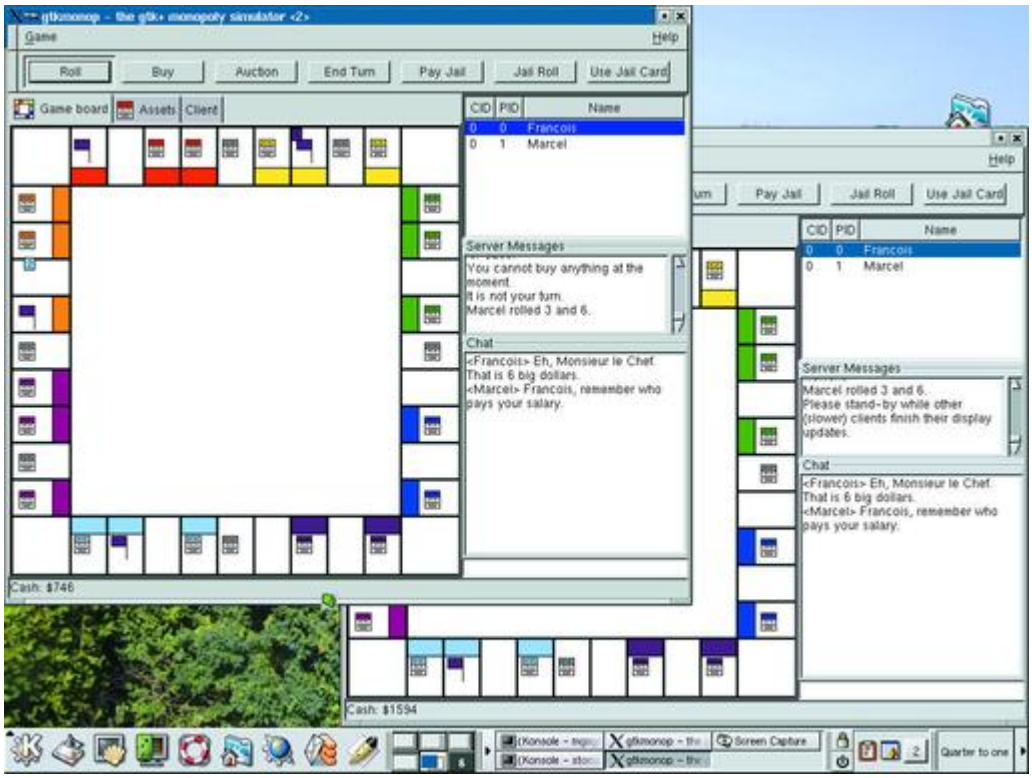


Figure 2. Squaring off for

In the Linux world, we sometimes joke about world domination, meaning (of course) Linux's domination of the server world, the desktop, the PDA and so on. It seems that for some of us, this obsession goes back to the days before Linux was born. Many years ago I whiled away a great deal of time with friends playing a little friendly game of “conquer the world”. I do not need to tell you the risk in such an enterprise. That, by the way, was also the game's name: *Risk*. I occasionally miss those days, but now I need no longer fear because our old friend, Ricardo Quesada, decided that he too missed those days. Ricardo is the author of *Tenes Empanadas Graciela* (or just plain *TEG*). Along with his dedicated band of developers and gamers, the classic world conquest board game lives.

```
tar -xzvf teg-0.8.0.tar.gz
cd teg-0.8.0
./configure
make
make install
```

In order to get a game going over your network, you must now run the tegserver program. Please note that you do not run this as the root user; in fact, the program will not allow this. With the server running, start another session, this time by running the tegclient program. You will be asked for a server to connect to and that will be (presumably) the name of the host on which you started the server.

Once you have connected to the server, you will be asked to choose your color (red, blue, pink, yellow, black or green). In order for a game to proceed, you

need at least two people connected and up to a maximum of six. When enough players have joined in, you can click Start to begin the game. This is a game of strategy, but the objective is simple: conquer the world. You'll notice, however, when you start the game, you also can run modified objectives, so-called secret missions. It also is possible to join in as an observer, watching others play the game.

*TEG* has three different map themes. These are classic, sentimental and my personal favorite, modern. Changing from one theme to another can be done during a running game by clicking on the Preferences tab. Figure 3 shows a game in progress between myself and my faithful waiter.

Figure 3. Marcel Regains Control of France

As with the other games I covered, *TEG* provides a chat window at the bottom of the screen so that you can send short messages to your fellow players during the course of a game. It is almost like sitting around the table at your grandparents' country home, *non*? The fun, the playful competition—it is all there, and now distance means nothing. So call up your cousin Henri in Versailles (or wherever), convince him to load Linux, and let the games begin.

*Oui*, François. I see the terrible clock and she is taunting us, *non*? *Mes amis*, I fear that closing time is upon us once again. When we cook with Linux, it seems that time vanishes. But do not worry. Finish your games. Drink your wine. François, I believe that I own this property. Prepare to pay up.

Until next time; *au revoir*, *mes amis*. Your table will be waiting. *A votre santé*! *Bon appétit*!

Resources

**Marcel Gagné** (mggagne@salmar.com) is president of Salmar Consulting, Inc., a systems integration and network consulting firm and the author of Linux System Administration: A User's Guide, published by Addison-Wesley.

Archive Index Issue Table of Contents

Advanced search

# Practical Threat Analysis and Risk Management

**Mick Bauer**

Issue #93, January 2002

Threat analysis won't make you sleep any better at night, but it will help ensure that the right things keep you awake.

If you've been reading this column awhile, you know I like to balance technical procedures, tools and techniques with enough background information to give them some context. Security is a big topic, and the only way to make sense of the myriad variables, technologies and black magic that figure into it is to try to understand some of the commonalities between security puzzles.

One piece common to each and every security scenario is the threat. Without a threat there's no need for security measures. But how much time do you spend identifying and evaluating threats to your systems, compared to the time you spend implementing and (I hope) maintaining specific security measures? Probably far too little time. If so, don't feel bad, even seasoned security consultants spend too little time on threat analysis.

This is not to say you need to spend hours and hours on it. My point is that, ideally, threats to the integrity and availability of your critical systems should be analyzed systematically and comprehensively; threats to less essential but still important systems at least should be thought about in an organized and objective way.

## Assets: You've Got 'em, the Bad Guys Want 'em

Before we dive into threat analysis, we need to cover some important terms and concepts. First, what does threat mean? Quite simply, a threat is the combination of an asset, a vulnerability and an attacker.

An asset is anything you wish to protect. In information security scenarios, an asset is usually data, a computer system or a network of computer systems. We want to protect those assets' integrity and, in the case of data, confidentiality.

Integrity is the absence of unauthorized changes. Unauthorized changes result in that computer's or data's integrity being compromised. This can mean that bogus data was inserted into the legitimate data, or parts of the legitimate data were deleted or changed. In the case of computers, it means that configuration files have been altered by attackers in such a way as to allow unauthorized users to use the system improperly.

We also want to protect the confidentiality of at least some of our data. This is a somewhat different problem than that of integrity, since confidentiality can be compromised completely passively. If someone alters your data, it's easy to detect and analyze by comparing the compromised data with the original data. If an attacker illicitly copies (steals) it, however, detection and damage-assessment is much harder since the data actually hasn't changed.

For example, suppose ABC Corporation has an SMTP gateway that processes their incoming e-mail. This SMTP gateway represents two assets. The first asset is the server itself, whose proper functioning is important to ABC Corp.'s daily business. In other words, ABC Corp. needs to protect the integrity of its SMTP server so its e-mail service isn't interrupted.

Secondly, that SMTP gateway is host to data contained in the e-mail that passes through it. If the gateway's system integrity is compromised, then confidential e-mail could be eavesdropped and important communications tampered with. Protecting the SMTP gateway, therefore, is also important in preserving the confidentiality and integrity of ABC Corp.'s e-mail data.

Step one in any threat analysis, then, is identifying which assets need to be protected and which qualities of those assets need protecting.

### Vulnerabilities

Step two is identifying known and plausible vulnerabilities in that asset and in the systems that directly interact with it. Known vulnerabilities, of course, are much easier to deal with than vulnerabilities that are purely speculative. (Or so you'd think, but an alarming number of computers connected to the Internet run default, unpatched operating systems and applications.) Regardless, you need to try to identify both.

Known vulnerabilities often are eliminated easily via software patches, careful configuration or instructions provided by vendor bulletins or public forums. Those that can't be mitigated so easily must be analyzed, weighed and either protected via external means (e.g., firewalls) or accepted as a cost of doing whatever it is that the software or system needs to do.

Unknown vulnerabilities by definition must be considered in a general sense, but that makes them no less significant. The easy way to illustrate this is with an example.

Let's return to ABC Corporation. Their e-mail administrator prefers to run sendmail on the ABC Corp.'s SMTP gateway because she's a sendmail expert, and it's done the job well for them so far. But she has no illusions about sendmail's security record; she stays abreast of all security bulletins and always applies patches and updates as soon as they come out. ABC Corp. is thus well protected from known sendmail vulnerabilities.

ABC's very hip e-mail administrator doesn't stop there, however. Although she's reasonably confident she's got sendmail securely patched and configured, she knows that buffer-overflow vulnerabilities have been a problem in the past, especially since sendmail is often run as root (i.e., hijacking a process running as root is equivalent to gaining root access).

Therefore, she runs sendmail in a “chroot jail” (a subset of the full filesystem) and as user “mail” rather than as root, employing sendmail's SafeFileEnvironment and RunAsUser processing options, respectively. In this way the SMTP gateway has some level of protection not only against known vulnerabilities, but also against unknown vulnerabilities that might cause sendmail to be compromised, but hopefully not in a way that causes the entire system to be compromised, too.

### Attackers

The last piece of the threat puzzle we'll discuss before plunging into threat analysis is the attacker. Attackers, also sometimes called “actors”, can range from the predictable (disgruntled ex-employees, mischievous youths) to the strange-but-true (drug cartels, government agencies, industrial spies). When you consider possible attackers, almost any type is possible; the challenge is to gauge which attackers are the most likely.

A good rule of thumb in identifying probable attackers is to consider the same suspects your physical security controls are designed to keep out, minus geographical limitations. This is a useful parallel: if you install an expensive lock on the door to your computer room, nobody will ask, “Do you really think the maintenance staff will steal these machines when we go home?”

Computer security is no different. While it's often tempting to say “my data isn't interesting; nobody would want to hack me”, you have no choice but to assume that if you're vulnerable to a certain kind of attack, some attacker eventually will probe for and exploit it, regardless of whether you're imaginative enough to

understand why. It's considerably less important to understand attackers than it is to identify and mitigate the vulnerabilities that can feasibly be attacked.

### Simple Risk Analysis: ALEs

Once you've compiled lists of assets and vulnerabilities (and considered likely attackers), the next step is to correlate and quantify them. One simple way to quantify risk is by calculating annualized loss expectancies (ALEs).

For each vulnerability associated with each asset, you estimate first the cost of replacing or restoring that asset (its single loss expectancy) and then the vulnerability's expected annual rate of occurrence. You then multiply these to obtain the vulnerability's annualized loss expectancy.

In other words, for each vulnerability we calculate: single loss expectancy (cost) × (expected) annual rate of occurrences = annualized loss expectancy.

For example, suppose Mommenpop, Inc., a small business, wishes to calculate the ALE for denial-of-service (DOS) attacks against their SMTP gateway. Suppose further that e-mail is a critical application for their business; their ten employees use e-mail to bill clients, provide work estimates to prospective customers and facilitate other critical business communications. However, networking is not their core business, so they depend on a local consulting firm for e-mail-server support.

Past outages, averaging one day in length, have tended to reduce productivity by about one-fourth, which translates to two hours per day per employee. Their fallback mechanism is a fax machine, but since they're located in a small town, this entails long-distance telephone calls and is expensive.

All this probably sounds more complicated than it is; it's much less imposing expressed in spreadsheet form (Figure 1).

| Item Description | Estimated Cost |
|---|---|
| Recovery: consulting time from 3rd–party firm (4 hrs @ $150) | $600.00 |
| Lost productivity (2 hours per 10 workers @ avg. $17.50/hr) | $350.00 |
| FAX paper, thermal (1 roll @ $16.00) | $16.00 |
| long–distance FAX transmissions (20 @ avg. 2 min @ $.25/min) | $10.00 |
| Total SLE for 1–day DOS attack against SMTP svr. | $950.00 |

Figure 1. Itemized Single Loss Expectancy

The next thing to estimate is this type of incident's expected annual occurrence (EAO). This is expressed as a number or fraction of incidents per year.

Continuing our example, suppose Mommenpop, Inc. hasn't been the target of espionage or other attacks by its competitors yet, and as far as you can tell, the most likely sources of DOS attacks on their mailserver are vandals, hoodlums, deranged people and other random strangers.

It seems reasonable to guess that such an attack is unlikely to occur more than once every two or three years; let's say two to be conservative. One incident every two years is an average of 0.5 incidents per year, for an EAO of 0.5. Let's plug this in to our ALE formula:

950 ($/incident) × 0.5 (incidents/yr) = 475 ($/yr).

The ALE for DOS attacks on Mommenpop's SMTP gateway is thus $475 per year.

Now suppose some vendors are trying to talk the company into replacing their homegrown Linux firewall with a commercial firewall; this product has a built-in SMTP proxy that will help minimize but not eliminate the SMTP gateway's exposure to DOS attacks. If that commercial product costs $5,000, even if its cost can be spread out over three years (to $2,166 per year after 10% annual interest), such a firewall upgrade would not appear to be justified by this single risk.

Figure 2 shows a more complete threat analysis for our hypothetical business' SMTP gateway, including not only the ALE we just calculated but also a number of others that address related assets, plus a variety of security goals.

| Asset | Security Goal | Vulnerability | SLE ($/incident) | ARO (incdts/yr) | ALE ($/yr) |
|---|---|---|---|---|---|
| SMTP Gateway | System Integrity | sendmail bugs | $2,400 | 0.5 | $1,200 |
| | | misc. system bugs | $2,400 | 0.5 | $1,200 |
| | System Availability | DOS Attacks | $950 | 0.5 | $475 |
| Confidential email (customer account info) | Data Confidentiality | Eavesdropping on Internet or ISP | $50,000 | 2 | $100,000 |
| | | Compromise of SMTP Gateway | $50,000 | 0.5 | $25,000 |
| | | Malicious insider | $150,000 | 0.33 | $49,500 |
| | Data Integrity | Forged email to/from customer | $10,000 | 1 | $10,000 |
| | | In-transit alteration on Internet or ISP | $10,000 | 0.25 | $2,500 |
| | | Compromise of SMTP Gateway | $10,000 | 0.5 | $5,000 |
| Non-confidential email (operations info) | Data Integrity | In-transit alteration on Internet or ISP | $3,000 | 0.25 | $750 |
| | | Compromise of SMTP Gateway | $3,000 | 0.5 | $1,500 |

Figure 2. Sample ALE-Based Threat Model

In this example analysis, customer data in the form of confidential e-mail is the most valuable asset at risk; if this is eavesdropped or tampered with, customers could be lost (due to losing confidence in Mommenpop), resulting in lost revenue. Different perceived potentials in these losses are reflected in the single loss expectancy figures for different vulnerabilities. Similarly, the different estimated annual rates of occurrence reflect the relative likelihood of each vulnerability actually being exploited.

Since the sample analysis in Figure 2 is in the form of a spreadsheet, it's easy to sort the rows arbitrarily. Figure 3 shows the same analysis sorted by vulnerability.

| Asset | Security Goal | Vulnerability | SLE ($/incident) | ARO (incdts/yr) | ALE ($/yr) |
|---|---|---|---|---|---|
| SMTP Gateway | System Integrity | sendmail bugs | $2,400 | 0.5 | $1,200 |
| SMTP Gateway | System Integrity | misc. system bugs | $2,400 | 0.5 | $1,200 |
| Confidential email (customer account info) | Data Confidentiality | Malicious insider | $150,000 | 0.33 | $49,500 |
| Confidential email (customer account info) | Data Integrity | In-transit alteration on Internet or ISP | $10,000 | 0.25 | $2,500 |
| Non-confidential email (operations info) | Data Integrity | In-transit alteration on Internet or ISP | $3,000 | 0.25 | $750 |
| Confidential email (customer account info) | Data Integrity | Forged email to/from customer | $10,000 | 1 | $10,000 |
| Confidential email (customer account info) | Data Confidentiality | Eavesdropping on Internet or ISP | $50,000 | 2 | $100,000 |
| SMTP Gateway | System Availability | DOS Attacks | $950 | 0.5 | $475 |
| Confidential email (customer account info) | Data Confidentiality | Compromise of SMTP Gateway | $50,000 | 0.5 | $25,000 |
| Confidential email (customer account info) | Data Integrity | Compromise of SMTP Gateway | $10,000 | 0.5 | $5,000 |
| Non-confidential email (operations info) | Data Integrity | Compromise of SMTP Gateway | $3,000 | 0.5 | $1,500 |

Figure 3. Same Analysis Sorted by Vulnerability

This is useful for adding up ALEs associated with the same vulnerability. For example, there are two ALEs associated with in-transit alteration of e-mail while it traverses the Internet or ISPs, at $2,500 and $750, for a combined ALE of $3,250. If a training consultant will, for $2,400, deliver three half-day seminars for the company's workers on how to use free GnuPG software to sign and encrypt documents, the trainer's fee will be justified by this vulnerability alone.

We also see some relationships between ALEs for different vulnerabilities. In Figure 3 we see that the bottom three ALEs all involve losses caused by the SMTP gateway's being compromised. In other words, not only will an SMTP gateway compromise result in lost productivity and expensive recovery time from consultants ($1,200 in either ALE, at the top of Figure 3), it will expose the

business to an additional $31,500 risk of e-mail data compromises, for a total ALE of $32,700.

Clearly, the ALE for e-mail eavesdropping or tampering caused by system compromise is high. Mommenpop, Inc. would be well-advised to call that $2,400 trainer immediately.

Problems with relying on the ALE as an analytical tool include its subjectivity (note how often in the example I used words like “unlikely” and “reasonable”) and, therefore, the fact that the experience and knowledge of whoever's calculating, rather than empirical data, ultimately determine its significance. Also, this method doesn't lend itself too well to correlating ALEs with each other (except in short lists as shown in Figures 2 and 3).

The ALE method's strengths, though, are its simplicity and its flexibility. Anyone sufficiently familiar with their own system architecture and operating costs, and possessing even a general sense of current trends in IS security (e.g., from reading CERT advisories and incident reports now and then), can create lengthy lists of itemized ALEs for their environment with little effort. If such a list takes the form of a spreadsheet, ongoing tweaking of its various cost and frequency estimates is especially easy.

Even given this method's inherent subjectivity (not completely avoidable in practical threat-analysis techniques), it's extremely useful as a tool for enumerating, quantifying and weighing risks. A well-constructed list of annualized loss expectancies can help you optimally focus your IT security expenditures on the threats likeliest to affect you in ways that matter.

## An Alternative: Schneier's Attack Tree Method

Bruce Schneier, author of *Applied Cryptography*, has proposed a different method for analyzing risk: attack trees. An attack tree, quite simply, is a visual representation of possible attacks against a given target. The attack goal (target) is called the root node; the various subgoals necessary to reach the goal are called leaf nodes.

To create an attack tree, you must first define the root node. For example, one attack objective might be “steal Mommenpop, Inc.'s customers' account data”. Direct means of achieving this could be 1) obtain backup tapes from Mommenpop's fileserver, 2) intercept e-mail between Mommenpop, Inc. and their customers and 3) compromise Mommenpop's fileserver over the Internet. These three subgoals are the leaf nodes immediately below our root node (Figure 4).

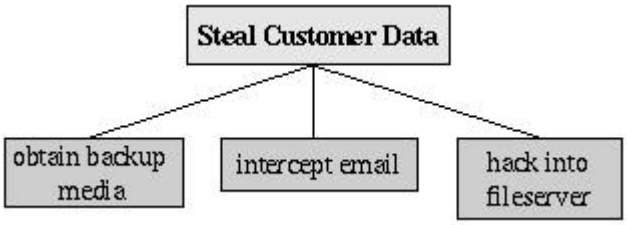



Figure 4. Root Node with Three Leaf Nodes

Next, for each leaf node you determine subgoals that will achieve that leaf node's goal, which become the next layer of leaf nodes. This step is repeated as necessary to achieve the level of detail and complexity with which you wish to examine the attack. Figure 5 shows a simple but more-or-less complete attack tree for Mommenpop, Inc.

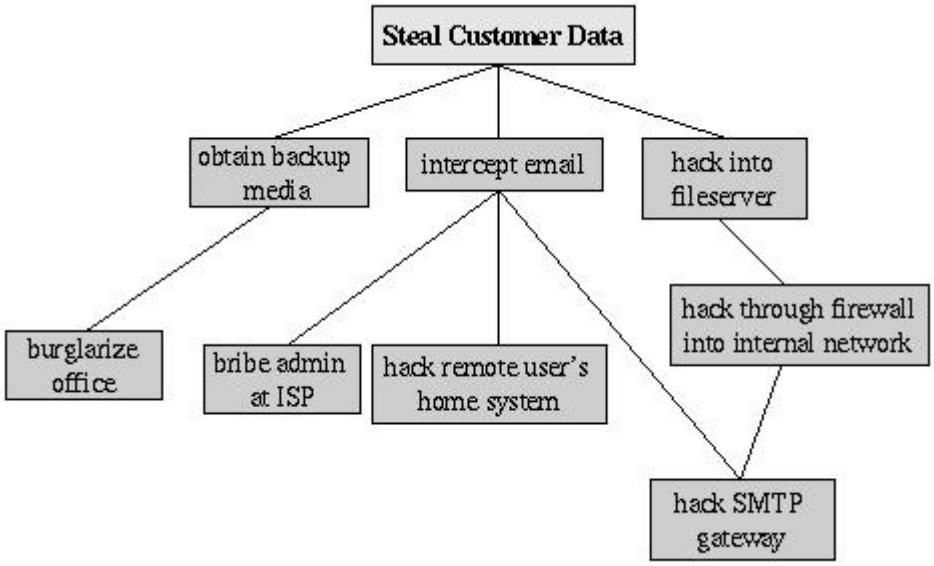



Figure 5. More-Detailed Attack Tree

No doubt you can think of additional plausible leaf nodes at the two layers shown in Figure 5 and additional layers as well. Suppose for our example, however, that this environment is well secured against internal threats (seldom the case), and that these are the most feasible avenues of attack for an outsider.

We see in this example that backup media are obtained most probably by breaking into the office; compromising the internal fileserver involves hacking in through a firewall, and there are three different avenues to obtain the data via intercepted e-mail. We also see that while compromising Mommenpop, Inc.'s SMTP server is the best way to attack the firewall, a more direct route would simply be to read e-mail passing through the compromised gateway.

This is extremely useful information; if this company is considering sinking more money into its firewall, it may decide that their money and time are

better spent securing their SMTP gateway. But as useful as it is to see the relationships between attack goals, we're not done with this tree yet.

After an attack tree has been mapped to the desired level of detail, you can start quantifying the leaf nodes. For example, you could attach a cost figure to each leaf node that represents your guess at the cost of achieving that leaf node's particular goal. By adding cost figures in each attack path, you can estimate relative costs of different attacks. Figure 6 shows our example attack tree with costs added (dotted lines indicate attack paths).

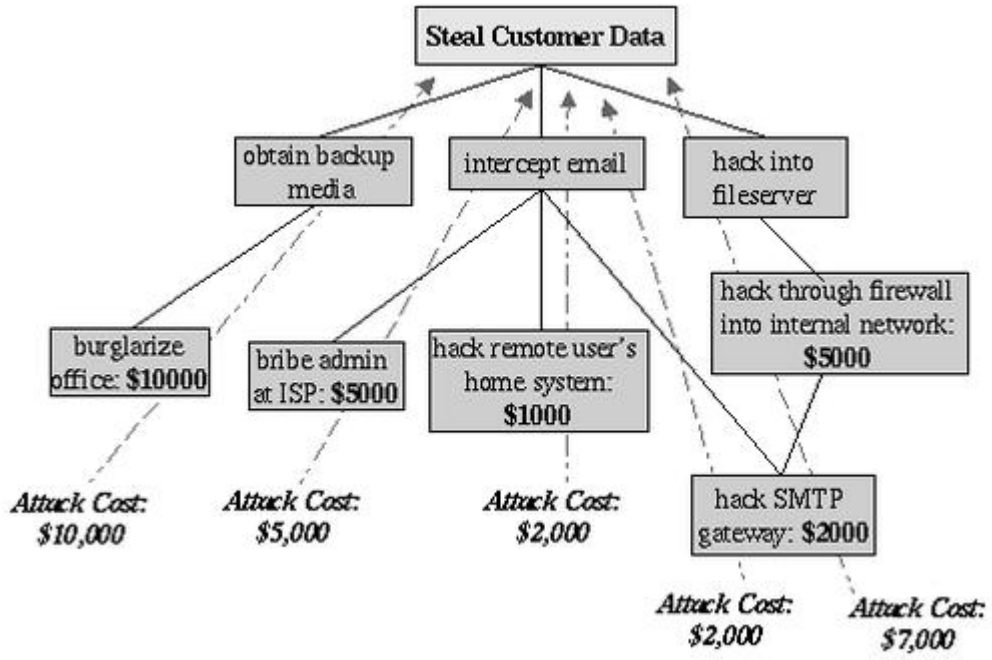



Figure 6. Attack Tree with Cost Estimates

In Figure 6 we've decided that burglary, with its risk of being caught and being sent to jail, is an expensive attack. Nobody will perform this task for you without demanding a significant sum. The same is true of bribing a system administrator at the ISP; even a corruptible ISP employee will be concerned about losing his or her job and getting a criminal record.

Hacking is a bit different, however. While still illegal, it's often perceived as being less risky than burglary. Furthermore, most organizations' computer defenses aren't nearly as difficult to breach as their physical defenses.

Having said that, hacking through a firewall takes more skill than the average script-kiddie possesses and will take some time and effort; therefore, this is an expensive goal. But hacking an SMTP gateway should be easier, and if one or more remote users can be identified, the chances are good that the user's home computer will be easy to compromise. Therefore, these two goals are much cheaper.

Based on the cost of hiring the right kind of criminals to perform these attacks, the most promising attacks in this example are hacking the SMTP gateway and hacking remote users. Mommenpop Inc., it seems, had better take a close look at their perimeter network architecture, SMTP server's system security and remote-access policies and practices.

Cost, by the way, is not the only type of value you can attach to leaf nodes. Boolean values such as feasible and not feasible can be used; a “not feasible” at any point on an attack path indicates that that entire path is infeasible. Alternatively, you can assign effort indices, measured in minutes or hours. In short, you can analyze the same attack tree in any number of ways, creating as detailed a picture of your vulnerabilities as you need to.

The cost estimates in Figure 6 are all based on the assumption that the attacker will need to hire others to carry out the various tasks. These costs might be computed very differently if the attackers themselves are skilled system crackers; in such a case time estimates for each node might be more useful than cost estimates.

## Defenses

The whole point of threat analysis is to try to determine what level of defenses are called for against the various things to which your systems seem vulnerable.

There are three general means of mitigating risk. Defenses can be categorized as means of reducing an asset's value to attackers mitigating specific vulnerabilities and neutralizing or preventing attacks.

Reducing an asset's value may seem like an unlikely goal, but the key is to reduce that asset's value to attackers, not to its rightful owners/users. The best example of this is encryption: all of the attacks described in the examples earlier in this article would be made irrelevant largely by proper use of e-mail encryption software.

If stolen e-mail is effectively encrypted, it can't be read easily by thieves. If it's digitally signed (also a function of e-mail encryption software), it can't be tampered with without the recipient's knowledge, regardless of whether it's encrypted too. A physical world example is dye-bombs: a bank robber who opens a bag of money only to see himself and his loot sprayed with permanent dye will have some difficulty spending that money. Asset-devaluation techniques like these don't stop attacks, but they have the potential to make them unrewarding and pointless.

Another strategy to defend information assets is to eliminate or mitigate vulnerabilities. Software patches are a good example of this: every single sendmail bug over the years has resulted in its developers distributing a patch that addresses that particular bug.

An even better example of mitigating software vulnerabilities is defensive coding. By running your source code through filters that parse, for say, improper bounds checking, you can help insure that your software isn't vulnerable to buffer-overflow attacks. This is far more useful than releasing the code without such checking and simply waiting for the bug reports to trickle back to you.

The defensive approach we tend to focus on the most (not that we should) is heading off attackers before they reach vulnerable systems. The most obvious example is firewalling; firewalls exist to stymie attackers. No firewall yet designed has any intelligence about specific vulnerabilities of the hosts it protects or of the value of data on those hosts. A firewall's function is to mediate all connections between trusted and untrusted hosts and minimize the number of attacks that succeed in reaching their intended targets.

Access control mechanisms such as username/password schemes, authentication tokens and smart cards also fall into this category since their purpose is to distinguish between trusted users and untrusted users (i.e., potential attackers). Note, however, that authentication mechanisms also can be used to mitigate specific vulnerabilities (e.g., using SecurID tokens to add a layer of authentication to a web application with inadequate access controls).

## Good-bye for Now

And with that, I bid you *adieu* for the next couple of months. Due to the demands of a book on Linux security I'm writing for O'Reilly & Associates, the Paranoid Penguin temporarily will be covered by others. Have no fear; they'll maintain the high level of paranoia and vigilance you've come to expect here. See you again in the April 2001 issue.

Resources

**Mick Bauer** (mick@visi.com) is a network security consultant in the Twin Cities area. He's been a Linux devotee since 1995 and an OpenBSD zealot since 1997, and enjoys getting these cutting-edge OSes to run on obsolete junk.

Archive Index Issue Table of Contents

Advanced search

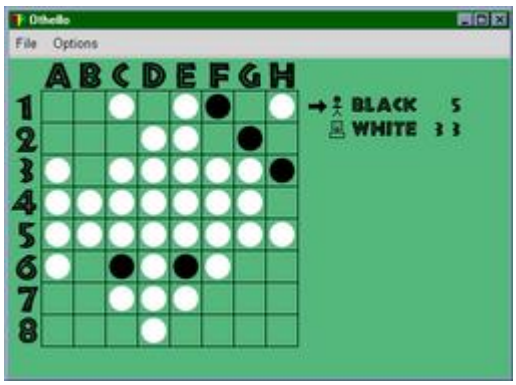
# Porting Gothello

**Robin Rowe**

Issue #93, January 2002

An exercise in porting Linux applications to Windows is made easier using GUI toolkits like GTK+.

Unfamiliar software has been a frequent excuse for Windows users not to switch to Linux. A lot of work is being done to undo this reasoning, not by bringing closed Microsoft software to Linux, but by bringing the benefits of open-source Linux software to Windows. Cross-platform versions of most popular Linux desktop applications are available today, including the GIMP image editor, StarOffice/OpenOffice office suite, the Mahogany e-mail client, the Amaya WYSIWYG Web Editor and many web browsers, including Mozilla and Opera. These all run on Linux and Windows, and some even support Macintosh.




Robin (black) being clobbered by the Windows version of Linux *Gothello* in AI play.

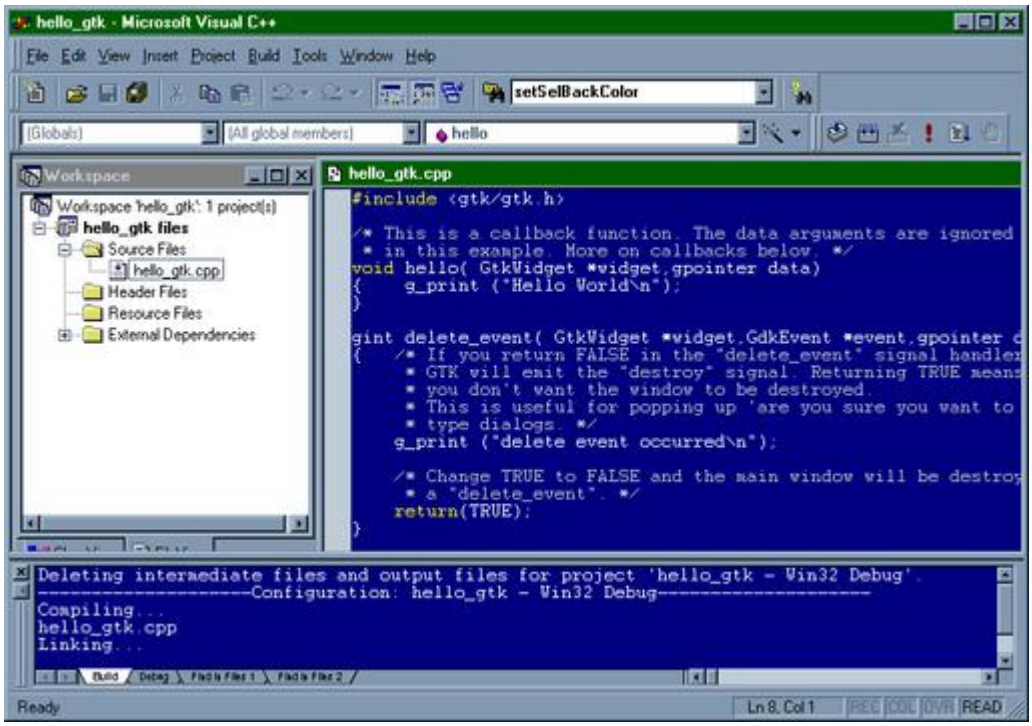
Popular server applications like Apache and MySQL also have been ported to Windows, but most programmers consider porting desktop graphical user interface (GUI) software to be the most daunting task. There are tricks that can help, but we'll show that it's not that hard to accomplish. In this article we'll port *Gothello*, a Linux version of the game *Othello*, to Windows. In the process we'll learn something about programming GTK+ (also called GTK). Our code changes to *Gothello* will be made in such a way as not to break existing Linux code.

*Othello*, a cross between checkers and tic-tac-toe, is a popular game. Released in Japan in 1971, *Othello* is a variant of the game *Reversi*, invented in England in 1888. Although many open-source software versions of *Othello* exist for Linux, none of them would compile under Windows until now.

A quick search for Linux applications at Freshmeat.net reveals many open-source, Linux-only versions of *Othello*: *Darwersi*, *Desdemona*, *Gothello*, *GReversi*, *QtHello*, *Rhino* and *xreversi*. Choosing software based on a GUI with cross-platform support makes porting much easier. Both GTK+ and Qt are Windows-ready, but X11 is not (eliminating *xreversi* from our list). Fortunately, most modern Linux software is based on those two portable GUI libraries that form the underpinnings of GNOME and KDE, respectively. For our first Linux-to-Windows 2000 porting project we chose GTK+-based *Gothello* written in C.

*Gothello* author Osku Salerma is a computer science student at the University of Helsinki, where Linus Torvalds studied while creating Linux. Salerma says, “Gothello came about because I was interested in game trees. The AI is implemented using Negamax search with Alpha-Beta pruning and Iterative Deepening.” *Gothello* uses AI to provide a virtual opponent in the game. Salerma plans to work in UNIX systems programming after he graduates in a year or so.

GTK+ (GIMP Toolkit) is a popular library for creating graphical user interfaces in C or C++. Under the LGPL you can develop open software, free software or commercial software using GTK+. Although originally written for developing GIMP, GTK+ is used in a large number of software projects, including GNOME. GTK+ is built on top of GDK (GIMP Drawing Kit), a portable wrapper for platform-specific windowing APIs like Xlib and Win32. The primary authors of GTK+ are Peter Mattis, Spencer Kimball and Josh MacDonald.



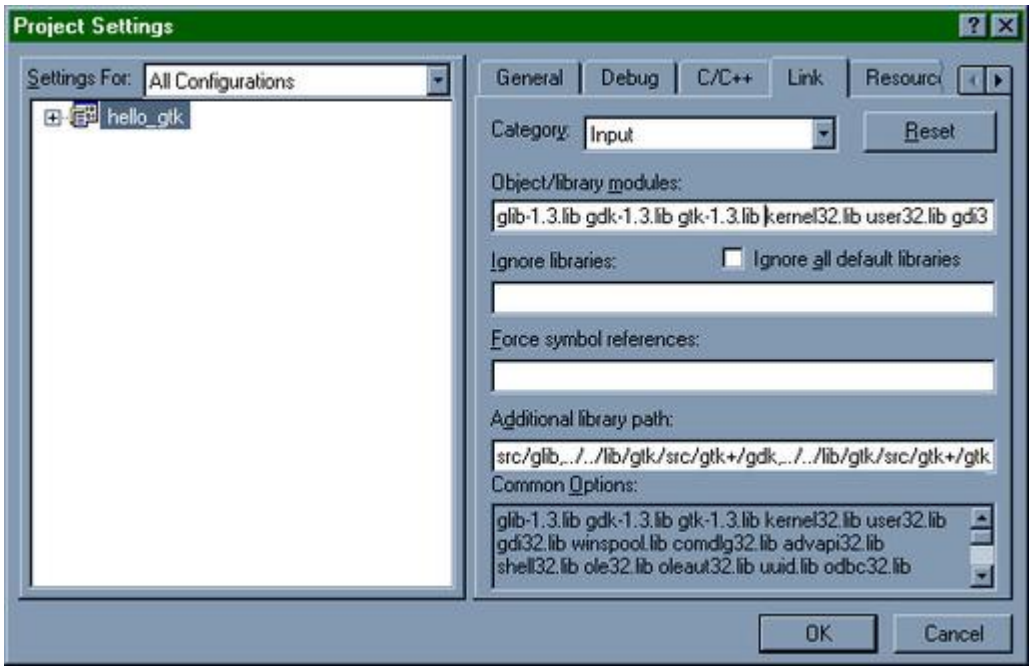
Visual C++ with Workspace, Source and Output Windows Visible

Tor Lillqvist, an engineer at Tellabs in Finland, ported GTK+ and GIMP to Windows in 1997. Lillqvist says he ported GIMP for fun so he could use it with his Minolta slide scanner in Windows. Although the current release of his Windows port, version 1.3.0, is from December 2000, there is a lively developers list in which Lillqvist actively takes part. “I am still working on it as much as time permits”, says Lillqvist. “I am the father of a three-month-old baby girl.”

Before attempting to build *Gothello* on Windows, we need to download and install the necessary GTK+ Windows libraries. A link to GTK+ for Win32 is provided on the main GTK+ web site. Download and unzip the glib, libiconv, gtk+ and extralibs files. The lib files are prebuilt, so there's no need to remake them. Where you place the files is a matter of personal preference, but we like /code/lib/gtk. We'll download *Gothello* and untar it in /code/oss/gothello. Note that Windows, except at the DOS command prompt, supports forward slashes in all file paths. Avoiding Windows-only back slashes saves trouble when switching back and forth with Linux.

Many books are available on GTK+ programming. We have *GTK+/Gnome Application Development* by Havoc Pennington (New Riders, ISBN 0-7357-0078-8). A book isn't required, however, because the GTK+ web site provides a nice on-line tutorial maintained by Tony Gale. We'll use his “Hello World” code example to test that we have GTK+ properly installed [Listing 1, available at ftp.linuxjournal.com/pub/lj/listings/issue93/5574.tgz]. The heavily commented code is a good introduction to how GTK+ works.

Open-source purists may prefer choosing an open-source Windows C++ compiler, and that is certainly feasible. Dev-C++ is a full-featured, open-source Windows integrated development environment (IDE) that we've used successfully in the past. This compiler incorporates the MinGW or Cygwin (gcc) compilers and the Insight or GDB debuggers. Most Windows C++ programmers, however, use the popular Microsoft Visual C++ compiler. It doesn't matter which Windows compiler is used, but we'll select Visual C++ so that we can cover some pitfalls that often trap less-experienced VC++ programmers. If you are a Linux open-source project leader accustomed to working with gcc, these VC++ tips can save you a lot of aggravation when working with the Windows VC++ programmers porting your code.



The deceptively simple Visual C++ project settings dialog, estimated to have more potential configurations than there are atoms in the universe.

Although VC++ can use Makefiles, smooth handling of project files is one of its most popular features. Project files are generated automatically by VC++. The typical VC++ programmer has no clue how to write a Makefile. Project files consist of a workspace file (extension .dsw) and one or more project files (extension .dsp). These text files are written by the {Project}{Settings} tabbed dialog box and are not intended for editing by hand—there's a zillion setting combinations. Not knowing what to look for here is one of the hardest aspects facing a user unfamiliar with VC++. A few settings are critical. Note that simply because a project will build doesn't mean the settings are right.

Now we come to the heart of setting up a complex VC++ project. Listen carefully because even though you may never use VC++ yourself, you almost certainly will have to tell any member of your team doing a Windows port how to set it up correctly. What VC++ programmers typically do is set absolute paths

to libraries in the project or configure their copy of VC++ to search implicitly for libraries where they have installed them. That would make the project files fail for anyone other than the person doing the port—not cool.

Here's how to set relative paths in the project from {Project}{Settings} in VC++:

```
{C/C++}{Preprocessor}{Additional include directories}:
        ../../lib/gtk/src/gtk+,../../lib/gtk/src/glib,
        ../../lib/gtk/src/gtk+/gdk
{C/C++}{Use runtime library}:
        Multi-threaded
{Link}{Input}{Object/library modules}:
        glib-1.3.lib gdk-1.3.lib gtk-1.3.lib
        (added before the others here)
{Link}{Input}{Additional library path}:
        ../../lib/gtk/src/glib,../../lib/gtk/src/gtk+
        /gdk,../../lib/gtk/src/gtk+/gtk
```

Make sure that All Configurations is selected before entering these settings, or else after you get the Debug version to build you'll have to do it over again with Release settings. Since GUI applications typically are multithreaded, it is always prudent to turn that on.

Our hello_gtk project will build now, but it won't run. A Windows error message complains that we haven't installed the GTK+ dll files (dynamic link libraries). Copy these to the Windows system32 folder, or better still, create a new folder for them and set Windows {Control Panel}{System}{Advanced}{Environmental Variables}{Path}. (Note: path changes won't take effect in the VC++ debugger without restarting VC++.)

After all this work to get VC++ and Windows configured properly, the *Gothello* porting effort itself isn't so hard. The first step is to deal with build errors from missing UNIX include files that don't exist under Windows, such as <sys/time.h> and <unistd.h>. Fortunately, most of the functions and type definitions included in these missing files are available somewhere in Windows. It's a bit of an Easter-egg hunt to find the right Windows header file. Some are documented and easily found, others are not. It turns out that <winsock.h> is a good place to look for undocumented, UNIX-compatible Windows calls. Berkeley sockets incorporate many common UNIX types and were implemented as part of Windows sockets.


Building and running the GTK+ “Hello World” test application proves that GTK+ is installed correctly.

Code revisions to *Gothello*'s timer.h include:

```
#ifdef _WIN32
#include <winsock.h>
#else
#include <sys/time.h>
#include <unistd.h>
#endif
```

Windows compilers implicitly define the system variable _WIN32. Using that name as a conditional is the standard method of making code automatically sense whether it is being compiled under Windows or some other operating system. This allows us to keep good Linux code in place while substituting Windows-specific code where needed.

Fixing *Gothello*'s timer.c is a little more difficult because we encounter a UNIX function not implemented anywhere in Windows:

```
#ifdef _WIN32
#include <sys/timeb.h>
#include <sys/types.h>
#include <winsock.h>
void gettimeofday(struct timeval* t,void* timezone)
{       struct _timeb timebuffer;
        _ftime( &timebuffer );
        t->tv_sec=timebuffer.time;
        t->tv_usec=1000*timebuffer.millitm;
}
#endif
```

Having a copy of *Using C on the UNIX System* by David Curry (O'Reilly & Associates, ISBN 0-937175-23-4) comes in handy to tell us what the gettimeofday() function is supposed to do.

Code revisions to *Gothello*'s child.c include:

```
#ifdef _WIN32
#include <stddef.h>
#include <io.h>
#include <stdlib.h>
#include <winsock.h>
#include <stdio.h>
#else
#include <sys/time.h>
#include <unistd.h>
#endif
```

Making child.c compile wasn't hard. We just had to track down the relevant Windows include files when the compiler complained about size_t, timeval, fd_set, FD_ZERO, FD_SET, tv_sec, tv_usec, select, NULL and _exit.

Code revisions to *Gothello*'s gtk_main.c include:

```
#ifdef _WIN32
#include <io.h>
#include <fcntl.h>
#include <process.h>
#else
#include <unistd.h>
#endif
```

It isn't until the body of gtk_main.c that we break a sweat. Windows doesn't fork(), and that can be a big problem in porting Linux applications. With fork(), Linux creates a clone of the running program in memory that can then branch to perform a simultaneous task. Windows has the less-powerful spawn() function but prefers to do its multitasking with threads. We'll rewire *Gothello*'s main() to make it think that a Windows thread is a child process. *Gothello* uses pipes to communicate between parent and child processes, a standard approach. Pipes work in Windows threads, too.

Explaining this thread (Windows) and fork (Linux) code in detail would be too tedious, but the design is based on the simple idea of creating two thread functions containing the code that the child and parent processes each would execute after forking. Be careful not to put the gtk_main() message pump loop function into these threads; doing so would stall the Windows message queue and lock up the application. If you are designing a Linux project with porting in mind, you can save yourself some trouble by using threads from the beginning. Linux threads are similar to those found in Windows.

The code in Listing 2 [available at ftp.linuxjournal.com/pub/lj/listings/issue93/5574.tgz] is much easier to understand if you read the functions in reverse order. The code in read_thread() requires some special explanation. Inside *Gothello* init_all() we make this minor modification, commenting out gdk_input_add() in Windows:

```
#ifndef _WIN32
    gdk_input_add(to_parent[0], GDK_INPUT_READ,
                  read_from_child, NULL);
#endif
```

When we first tried to run *Gothello* under Windows, the program locked up. The Windows port of GTK+ doesn't seem to be happy with gdk_input_add(). Threads are a better way to go than multitasking in the message loop. Instead of using **gdk_input_add()** to detect input on a pipe inside the GTK+ message queue, we handle that as an independently running, simultaneous thread. We also make a minor change to child.c to fake out the select() call that in Linux sets the pipe to operate asynchronously:

```
#ifdef _WIN32
    ret = 0;
#else
    ret = select(to_child + 1, &set, NULL, NULL, &tv);
#endif
```

Porting *Gothello* was our first time using GTK+. It would have been easier to do the port had we first built *Gothello* under Linux. We didn't do that because we wanted to evaluate how hard it would be for a Windows programmer unfamiliar with GTK+ to do a port sight unseen. It took myself and partner Gabrielle Pantera a couple of weeks, working part-time during nights and

weekends. That popular Linux GUI toolkits such as GTK+ and Qt are portable vastly simplifies the task of making Linux applications run on Windows.

Ported Linux desktop GUI applications can ease the learning curve of Windows users considering a move to Linux. And Linux users can access their favorite applications when using a Windows machine that may be at the office or that belongs to a friend. You even can program enhancements to your favorite open-source application for Linux while working in another operating system. Besides, there's nothing wrong with easing a million Windows C++ programmers into open-source programming.

Resources

email: Robin.Rowe@MovieEditor.com

**Robin Rowe** (robin.rowe@movieeditor.com) is a partner in MovieEditor.com, a technology company that creates internet and broadcast video applications. He has written for *Dr. Dobb's Journal*, the *C++ Report*, the *C/C++ Users Journal* and *Data Based Advisor*.

Archive Index Issue Table of Contents

Advanced search

LINUX JOURNAL



# Network Abuse

**David A. Bandel**

Issue #93, January 2002

On network basics and revisiting programs previously featured.

As Linux (ab)users, most of us take networking for granted. Luckily for us, this is made easy due to the Linux kernel design. But this can be abused, and badly. Although UNIX and Linux administrators tend to have a better handle on networking than most others, I want to relate something that, while it happened on a Windows network and was exacerbated by poor practice, also can happen on a Linux network.

Some 18 months or more ago, I reviewed a client's network before installing a Linux firewall. Their network was running 10Mb and had some long cable runs (in some places using telephone cable rather than Cat 5 cable), poorly terminated cables and up to four hubs cascaded. They also were running three protocols (IP, IPX and NetBEUI). Needless to say, with over 15% packet loss, communications were poor. A couple of weeks ago, this client decided to upgrade from a slow 64k frame-relay connection to a significantly faster 128k wireless connection. At over 26% packet loss, their network collapsed. They're doing better today with a redesigned network: 100Mb circuits, one protocol (IP), the removal of the hub cascade and correct, well-made cables. But their administrators did not understand what had happened. We might blame this on the fact they were MCSEs and had no UNIX network training, but it can happen anywhere administrators don't have a full grasp of network basics. There's a lot of networks out there, and many are in bad shape. So don't laugh, you might just inherit one.

Before I get started reviewing software, I want to note that this is the first issue of the fourth year of Focus on Software. During this time, some programs featured here have advanced significantly, while others have (seemingly) disappeared. So I'm going to feature one program from three years ago each month; if you have a favorite I featured in the past, let me know.

For this month's flashback, I looked at several very good programs, including the GTK+ Equation Grapher (geg), gtkfind (which seems to have disappeared from the Web) and X Northern Captain, among others, but my selection is PySol.

PySol http://www.oberhumer.com/opensource/pysol/

Whenever I upgrade my system, I always try to clean out all the cruft (and I have a *lot* of cruft). Well, the day I upgraded my system, not less than two members of the family complained that PySol was gone. Few programs are as used as this one, so despite the fact that it's a game, this one deserves another mention. I called it “Windows Solitaire on steroids” three years ago, but it has really advanced—sound, music, card sets and hundreds of games. This eclipses any commercial card game software I've seen. Requires: Python.

Netdude netdude.sourceforge.net

Hard-core network gurus might like to plow through a tcpdump file and find it easy to read. But if you're just starting out, Netdude is a very nice utility that will read a tcpdump file and format the output so that it's extremely readable. You can even make changes to the file and save it back. Requires: libgtk, libgdk, libgmodule, libdl, libXext, libX11, libm, libglib, libpcap, glibc.

ifmonitor ifmonitor.preteritoimperfeito.com

The ifmonitor utility will watch an interface for you, collect data on it from the /proc filesystem and insert it into an SQL database. A PHP script is then available to access that data and display it as a graph in a web browser. It's simple and easy to install and use. Requires: MySQL, /proc, Perl, Perl module DBD::MySQL, PHP with gd and MySQL, web server with PHP, web browser.

GtkBalls gtkballs.antex.ru

This should keep folks occupied for some time. Try to align five same-color balls to remove them and score points. Each time you don't, three new balls appear randomly on the grid. Requires: libtk, libgdk, libgmodule, libglib, libdl, libXext, libX11, libm, glibc.

Manhattan Virtual Classroom manhattan.sourceforge.net

This is an extremely simple, easy-to-use system for students and teachers. A truly virtual classroom, the author built it with security in mind. While not the simplest of applications to install, the author provides clear, concise installation instructions. Follow them to the letter, and you can't go wrong. If you are a

teacher or consultant working with a school, this program deserves a demonstration. Requires: glibc, Apache Web Server.

Celestia www.shatters.net/celestia

This is an extremely impressive 3-D star viewer. You can visit the known universe from your computer. The graphics are well done, and you have a lot of data available. While the beauty is striking, what it will do to your system is just as striking. I may not have the latest and greatest gigahertz system going, but I didn't think it was that slow until I ran this. I wouldn't even consider trying to run this on a classic Pentium I. Requires: libpng, libjpeg, libGLU, libGL, libSM, libICE, libXmu, libXi, libXext, libX11, libstdc++, libm, libz, libpthread, libdl, libXt, glibc.

Until next month.

email: david@pananix.com

**David A. Bandel** (dbandel@pananix.com) is a Linux/UNIX consultant currently living in the Republic of Panama. He is coauthor of Que Special Edition: Using Caldera OpenLinux.

Archive Index Issue Table of Contents

Advanced search

# Take Linux with You Wherever You Go

**Rick Lehrbaum**

Issue #93, January 2002

Rick takes a look at a pair of recent announcements ofinteresting new gadgets with Linux embedded inside.

GITWiT (Kirkland, Washington) is currently developing a Linux-based wireless handset that combines communications and entertainment within a uniquely customizable two-part design. The company plans to aim the device at the lucrative teen market initially by offering a colorful phone with cool teen-oriented capabilities.

According to GITWiT's market research, roughly 25% of the 40 million teens in the US own wireless phones today, leaving 30 million potential additional subscribers up for grabs.

## The How and Why of a Two-Part Phone Design

Physically, the GITWiT phone consists of two pieces (see photo). The GITWiT-enabled handset contains the core telephony functionality. A software-enabled cover, called a Smart Skin, snaps onto the handset core to define a specific and easily alterable look, feel and wireless-user experience.

GITWiT Phone's Two-Part Design

Each Smart Skin contains a Smart Key, which is built around an industry standard, secure smart card module. Smart Keys contain encrypted data and software that changes the wireless-user experience to match the theme of the skin.

As an example, users snapping on skins branded by their favorite music group might get a new look and feel for the user interface, plus customized ring tones and graphics, preconfigured wireless web bookmarks and subscription content for the group's latest CDs and upcoming concerts.

GITWiT expects its innovative approach to be popular with wireless carriers because it permits them to embrace changing fashion and entertainment trends rapidly by refreshing the relatively expensive core cellular component with an inexpensive snap-on Smart Skin.

## Under the Hood

From a technical perspective, GITWiT-enabled handsets are composed of three main parts: a cellular processor, a proprietary Smart Key operating system and an embedded host processor.

The host processor, an embedded computer running Linux, is the heart of the phone. It is built around an ARM7 microprocessor and contains a number of embedded peripherals and I/O interfaces for the keypad, color LCD and for communicating with the Smart Skins. A separate cellular processor handles all of the on-air cellular communications.

GITWiT uses the 2.4.5 Linux kernel with ARM patches from Russell King and Nicolas Pitre, as well as a number of in-house patches. The system software also includes portions of Microwindows for its GUI, as well as BusyBox.

## Why Linux?

“We chose Linux for a number of reasons”, explains GITWiT VP of Engineering Peter Zatloukal. He continues:

> We are building a user interface that is leagues beyond what exists on current wireless phones, and Linux provides us with a rich environment with which to render our ideas.
>
> Also, since most of our development work is in the application and object layers, we're glad to be using an open-source kernel so that we can contribute to

> efforts that make it easier for others to develop embedded solutions.
>
> We feel that increasing the ease of innovation in the embedded space, even if it also strengthens our competitors, helps us because it broadens the whole market. We like the world where low barriers to entry fuel competition around what really matters—a richer wireless-user experience.

See www.GITWiT.com for more information.

### IBM and Citizen Watch Develop Linux-Based WatchPad

IBM Research and Citizen Watch have announced a collaboration through which Linux-based WatchPad prototypes and related technologies are being developed. The collaborative project, which builds on IBM's earlier Linux Watch efforts, has the goal of exploring a new type of personal information access devices for the pervasive computing era. IBM Research first demonstrated the Linux Watch last year, in an effort to illustrate the viability of Linux across all platforms, from the S/390 to the smallest intelligent devices.


The IBM/Citizen WatchPad

Citizen Watch decided to work with IBM to develop enhanced features and new technologies for use in future “intelligent watches”, such as using them as communication devices. Thus far, Citizen Watch has contributed packaging and component design, including display and input device. IBM provided the hardware architecture, system design and software—including Linux. The two companies also say they plan to collaborate with universities by sharing the WatchPad technology for joint research, in hopes of accelerating progress in the development of next-generation intelligent devices.

The WatchPad contains a high-speed, low-power 32-bit MPU, 16MB of Flash memory and a quarter-VGA (320 × 240 pixels) LCD. Wireless connectivity includes both Bluetooth and infrared. Users interact with the device through a touch panel, buttons and modified winding knob. In addition, an accelerometer is embedded in the device to explore the possibility of arm movement being used as an input method.

Here are a few key technical specs of the WatchPad, taken from a fact sheet distributed by IBM Research:

Hardware

- Size: 65mm × 46mm × 16mm
- Weight: 43g (without wrist band)
- CPU: high-speed, low-power 32-bit MPU (18-74MHz)
- Input devices: touch panel, a winding crown switch, button
- Display: 320 × 240 dots, monochrome liquid crystal display
- Memory: 8MB low-power DRAM, 16MB Flash
- Interfaces: Bluetooth wireless technology (v1.1, voice-enabled), IrDA (V1.2), RS-232C (via a cradle)
- Others: speaker, microphone, vibrator, fingerprint sensor, accelerator sensor
- Power: Li-Ion battery
- Cradle: RS-232C, AC adaptor and AA batteries

Software

- Operating system: Linux kernel version 2.4
- GUI: Microwindows
- Bluetooth stack: IBM BlueDrekar (L2CAP, SDP, RFCOMM)

**News Flash!**

Sharp Electronics is taking orders for their Zaurus SL-5000D (developer edition) Linux/Java PDA from developers beginning in November 2001. The price for the developer model is $399 US, and it includes 32MB DRAM and 16MB Flash.

The device is based on a 206MHz Intel StrongARM system-on-chip processor, and it has a 3.5" 240 × 320-dot pixel (quarter VGA) reflective TFT 65,536 color LCD with touch-panel support and, of course, runs an embedded Linux operating system. The software stack is based on Lineo's Embedix, Trolltech's Qt Palmtop Environment, Opera's web browser and a PersonalJava v1.2-compliant Java runtime environment. Two expansion slots are provided: a secure digital (SD) card slot, used mainly for Flash memory, plus a CompactFlash slot used for communications interfaces, a digital camera attachment, Flash memory and more.

A unique and highly desirable feature of the SL-5000D is its full QWERTY keyboard, which is accessed by sliding the bottom portion of the device downward. See developer.sharpsec.com for more information.

Zaurus SL-5000D Linux/Java PDA

**Rick Lehrbaum** (rick@linuxdevices.com) created the LinuxDevices.com “embedded Linux portal”. Rick has worked in the field of embedded systems since 1979. He cofounded Ampro Computers, founded the PC/104

> Consortium and was instrumental in creating and launching the Embedded Linux Consortium.

email: rick@linuxdevices.com

Archive Index Issue Table of Contents

Advanced search

# Open Source Radio

**Doc Searls**

Issue #93, January 2002

An interview with Wild Bill Goldsmith of KPIG and Radio Paradise.

KPIG is an anomaly. It's a top-rated commercial radio station programmed entirely by its own DJs. It's a serious radio station with a terrific sense of humor. It's as much a fixture in its community as a statue in front of a courthouse—and a lot more fun.

The station's format has been described as “mutant cowboy rock and roll”, but it's hard to make any label stick on a station that is more real, more fun and more like-it-oughta-be than (by my conservative estimate) all the other commercial music stations in the country put together. Its roots go back to the legendary KFAT, and beyond that to the KRAB Nebula of loosely affiliated noncommercial stations out of which “community radio” and “free form radio” evolved in the sixties. It's a history that's as old and woolly as UNIX's. The resemblance doesn't end there, either.

The words free and open are taken seriously by KPIG and its faithful, who contribute code in the form of original voices, programming ideas and even musical selections. KPIG is fortuitously located in (no kidding) Freedom, California, and has one of the worst signals in the Salinas-Monterey-Santa Cruz market. If you're in the bowl of mountains that surround Monterey Bay, chances are you can get it. If you're not, your only choice is to punch up www.kpig.com and choose among a wide selection of streams, including a 128Mb MP3 spigot that's one of the prettiest-sounding signals that ever poured out of your speakers.

KPIG was the first commercial radio station to broadcast on the Web, and it has blazed trails ever since. This last spring, when AFTRA decided its commercial talent should be paid as much as 300% extra for commercials going out over the Net, thousands of commercial stations shut down their streams. KPIG just hacked around it. When the national ads come on, webstream listeners get pleasant filler.

To the constant astonishment of everybody but the station and its listeners, KPIG consistently manages to sit near the top of the local Arbitron ratings, despite its second-rate signal. It is, by all measures, a success. It is also full of open-source technologies that are in a good position to burn down “broadcasting as usual if the right hackers and serious radio fanatics get their hands on it. That's the idea behind what you're reading right now.

KPIG's hacker in chief is ”Wild Bill“ Goldsmith, a KPIG veteran going back to KFAT. He's now living full-time in (no kidding) Paradise, California, where he has steadily improved his own solo effort, Radio Paradise, alongside KPIG. I decided to get in touch with Bill after I received e-mail raves about Radio Paradise almost simultaneously from friends in Seattle, New York and North Carolina.

The sources were old radio pros who have since moved on to other professions but know the real thing when they hear it.

It was clear from a quick look at both KPIG and Radio Paradise that the stations were up to something more than radically good radio programming. Technically, both were easily customizable hacks built on open-source software and generic hardware. Specifically, Bill had put together what appeared to be a tightly integrated system that involved the music library, the web site, the whole audio chain, accounting, operations and scriptable tie-ins to potentially money-making partnerships with anybody who wanted to cross-promote anything, including stores, artists, labels or whatever—in a way that allowed as much live operation or automation as one saw fit.

This appeals to me. I'm a radio freak going back to my childhood, when I was a ham radio operator (even today the only code I know is Morse). My nickname, Doc, is the fossil remnant of the name I used on the air when I worked at WDBS, a KFAT-like station in North Carolina back in the seventies. Commercial radio today is a snatched body that bears only superficial resemblance to anything anybody loved in its golden age. Revitalizing radio is also a passion I share with Phil Hughes, *Linux Journal*'s chief geek and publisher, an old radio hand who fondly recalls KRAB's golden days in Seattle.

When I wrote and asked Bill exactly what he was hacking together at the two stations, he wrote this back:

> It's based on a set of software tools—for picking and scheduling music and doing voice tracks from anywhere over the Net, and for accepting and organizing listener feedback on my playlist. Everything I'm doing software-wise is 100% open source: Linux, PHP, Perl, Postgres and Icecast.
>
> I am convinced that what you see at www.radioparadise.com represents the future of radio, or of quality radio, anyway: very interactive, tightly controlled artistically (no random segues, everything happens for a reason), completely free from the influences of the radio/music industry hype machine (to the best of my ability, anyway) and supported primarily by voluntary contributions from listeners.
>
> This isn't a game plan that's going to make anyone rich. But it can make it possible for anyone with talent to make a very comfortable living without compromising their integrity in any way—and that's all I for one have ever wanted.

I suggested an interview and he agreed. We talked in late October, right after he finished putting up the latest KPIG web site.

**Doc** Hey, I like the new site. The Webcast links and the current song list are right up front. And I like the webloggy thing going on in the center of the page too. Makes it all very live. Very radio.

**Bill** It should be more webloggy-looking very soon, after the staff at the station takes it over.

**Doc** I've thought for a long time that the Linux and open-source development community could do something fun to blow commercial radio out of the water. Now it looks to me like you're doing the pioneering work here. Are your tracks in the snow the ones we should follow?

**Bill** I really think so. What I'm doing with the stations I'm working with is gradually evolving toward a complete open-source package for programming, managing and scheduling a radio station, with an integrated web site to go along with it. Basically everything that makes it possible to do what you see on the screen there at KPIG.com is all done with open-source stuff. There is not one lick of proprietary software in there anywhere.

**Doc** Give me a rundown of what I'm looking at when I visit the KPIG site. How is the content organized and served up here?

**Bill** Basically what builds the page is Apache running on a Linux box. It pulls a lot of stuff dynamically out of an SQL database in Postgres. And it also pulls in a lot of bits and pieces—little snippets of HTML that are built by scripts. An example of that is the Now Playing box. What you see there is a snippet of code that actually is written by the automation system that the DJs are using to choose and play back the music, which is another Linux server running the same kind of software. The DJ is looking at a private web page, by which they control what goes out on the stream.

**Doc** In other words, the disc jockey has an internal web page that works as a control console. It's interactive in some way.

**Bill** Yes. They use that to pick and schedule the music. The screen looks like any other radio station automation system. It doesn't even look like a web page. Most of them don't even know that they're looking at a web page. I run it in full-screen mode. They use this screen to control the server that does the audio playback, which they can program in advance. They can preview what all the segues are going to sound like, right down to the nth degree. And then they can walk away. You could—though we never would—run a completely automated

station this way. The capabilities are there. I use it that way for Radio Paradise and for Smooth Jazz (www.smoothjazz.com).

**Doc** Smooth Jazz is yours too?

**Bill** The technology of it or the back end: the automation and the piece that builds the web site.

**Doc** So this back end is writing scripts all over the place?

**Bill** Okay, when the DJ hits the button to start something, or the system starts something on its own because it's been programmed to do that by the DJ, it not only starts the music playback but also writes a new little Now Playing page. That song in the Now Playing slot moves down to the top of the next three. A total of four songs are listed: the song now playing and the most recent three. Each has its own page. The system also writes the page that you see when you click on Playlist, which is a history of what's been playing for the last six hours. Those actually are done on a separate machine and then FTPed over to the web server and included in the main page.

**Doc** Does all the music have to be on a hard drive somewhere?

**Bill** No. Most of it will be, but there are exceptions. At KPIG if the jock wants to play something not in the database, he can type in the title so the system will display it.

**Doc** And everything you're hearing on KPIG is MP3?

**Bill** They're high band rate MP3s. Mostly 256Kb. Some are 192. All are played out of a Linux box using a $90 sound card. The whole hardware cost for one of these operations is about $600. It's really cheap, which is why a business can be built around it. But it would be a specialized business: serious next-generation radio.

**Doc** I've been amazed at how little the public and noncommercial stations use MP3 as an encoding and transmission system on the Web. Most of them use Real and a few use Windows Media Player. Yet that seems like it would be more expensive to me. Is it just because they don't know better?

**Bill** Yeah.

**Doc** It's just branding.

**Bill** It's probably 80-90% branding. Setting up a free Real server is easy for somebody to do if they're not interested in serving more than 20 people at a

time. But there's nothing serious going on there. Okay, say 95% branding. And inertia. A lot of people started with Real back when it was the best choice and have stuck with it because they've had no real reason to switch. But man, if I was paying for a server license for a Real server, I'd drop that for MP3 or QuickTime or something else that was free. It's nearly impossible to recoup even the bandwidth costs downstream.

**Doc** Is there an advantage to Real at any of the different speeds?

**Bill** Real's latest codecs are much more optimized for low bit rates than MP3 is. They do have a quality advantage at bit rates below about 32k. That's the 5%. But the difference isn't that extreme.

**Doc** Is your system in tight enough shape for you to productize it?

**Bill** I would like to get this out there. I don't want to sell it as a product. I want it to be the nucleus of an open-source project. I don't want to be the Linus of the system, rather I want to use the system. That's why I built it in the first place. I'm not a programmer by trade. I've learned enough to build this—it's a lot of stuff, and I'm happy I know it—but it's a little too big for me to handle alone. Others need to come in and run with it.

**Doc** Have you looked into putting this up on SourceForge?

**Bill** I'm looking at a number of options. Right now I'd like somebody to come in from the outside and check out what we've got here. It's a really big project. It encompasses all these different activities involved in running a broadcast operation. It's not a tool for somebody who wants to do radio as a hobby. It's a tool for people who are serious about doing high-quality radio by using 21st-century tools.

**Doc** How many people know what high-quality radio is?

**Bill** In the industry, hardly anybody. All the people who would be open to doing what I'm talking about here have long since been driven out of the industry. KPIG is the exception. If I weren't hooked up with KPIG, I wouldn't have anything to do with commercial radio. I'd be off doing my own thing.

**Doc** KPIG breaks the mold in a lot of ways.

**Bill** Our staff is huge by medium-market radio standards. We have live DJs 24 hours a day. Nobody has that. Even in San Francisco you don't see that. Not anymore.

**Doc** You stream a wide selection, but you're basically about MP3s. Is that because just about everybody with a computer and a love of music has an MP3 player of some kind?

**Bill** The penetration isn't quite as high as for Real or Windows Media Player. But Real does a fine job of playing MP3 streams, too. In any case, the number of people out there who have Real, or Winamp or iTunes is pretty high.

**Doc** We're reviewing Apple's OS X here, which sits on Darwin, a form of BSD. And the default MP3 player shipping with it is iTunes, which comes with a tuner that demonstrates how hard it is to categorize stations. It's also fed by something called the Kerbango database. (Kerbango is the late embedded Linux radio company that was absorbed by 3Com and killed off early this year.)

**Bill** Apple hired a friend of mine who was a former Kerbango employee to maintain the iTunes database as a part-time project. He has KPIG under ”Americana“ and Radio Paradise under ”Alt/Modern Rock“. It used to be under ”Americana“, but I told him to change it.

**Doc** Who is going to be doing serious radio and is there a business in it?

**Bill** I think this is a nice small-scale business for one person who has a wide-ranging skill set—or for a small group of people. If it's done right, an on-line radio station at this point in time could probably support a staff of two or three people working full-time.

**Doc** What's the revenue model?

**Bill** At this point the major revenue source for all the successful operations I know about is listener donations.

**Doc** The public radio model.

**Bill** Yes. And it's much more applicable to this situation than the commercial radio model, which any number of people have tried to come in and make work on-line and failed at. As long as there is commercial-free competition, there is no reason anyone will want to listen to something that has advertising in it, unless it was spectacularly better than the alternative. And nobody's coming in and doing things that are spectacularly better.

**Doc** Seems to me the best on-line stations, like the best over-the-air stations—few as there are—sound like somebody who knows their stuff, playing their record collection for you.

**Bill** Which is the old underground FM radio system.

**Doc** Not many people have observed that the fundamental flaw with commercial radio is that its customers and consumers are different populations. That's a disadvantage noncommercial broadcasting does not have. The listeners are the customers. And on the Net it's easy to put out a pay jar so people can compensate the broadcaster for the service. Can't do that with a car radio.

**Bill** That payment system seems to work remarkably well at Radio Paradise.

**Doc** How?

**Bill** Here's my experience. For about a year or so people would occasionally write me and say "Hey, I'd like to pay for this. Do you accept donations?" And I'd always write back and say, "No, no. Keep your money." Until one day my wife and I were sitting around trying to figure how the hell we're going to pay for all this. And we said, "Y'know, people have been offering to give us money. Wonder what would happen if we made that possible?" So we put up the link and did a very, very low-key promotion—maybe once or twice a day on the stream—and a little blurb in the news section of the web site and a little link down at the bottom of the page. In the first month we got about $2,000 in donations. Which is about $2,000 more than we made the month before. Now we've got a target of $3,000, and we get about $3,000-$3,500 every month, plus another $300-$400 in affiliate program stuff, like with CDNOW.

**Doc** Since when?

**Bill** May.

**Doc** And that covers you?

**Bill** That's enough so I can drop a number of consulting projects I had been doing.

**Doc** Not bad for being this early in the game.

**Bill** Yeah. The station is more or less playing for itself.

**Doc** Will your expenses go up?

**Bill** We've got a free bandwidth deal going right now. After that ends the expenses will probably double, but I think the revenue will double as well. The audience is growing.

**Doc** Do you collect data on listeners?

**Bill** We're starting at KPIG. It's all voluntary for the listeners. The sense is that about 80% or better are at work. A lot of them are home workers with cable or DSL accounts. At work is also where people are looking for some music in the background and they've got a computer sitting there. They're also mostly in the US, about 80% or so. Quite a few in Europe. Age range is all over the place, but mostly in the 30s and 40s.

**Doc** What do you make of Live365, which hosts about 35,000 MP3 streams? It's like this vast mutant thing that manages to crash for me on three different platforms.

**Bill** They require that you use this little pop-up player of theirs. A little JavaScript thing.

**Doc** But do they demonstrate a business in brokering individual MP3 streams?

**Bill** I think there is definitely a market in doing infrastructure for on-line radio. That's basically what Live365 does. I question their revenue model, which is advertising. They seem to do almost random in-stream ad insertions. Their original model, like everyone else's, was to get rich selling banner ads on their web site once it had a gazillion visitors. We all know how well that worked.

**Doc** We all could have saved ourselves a lot of trouble if we read the MUTE buttons on our remote controls.

**Bill** [Laughter.] Live365 does have new people paying to stream through the service. But they also have thousands and thousands grandfathered in for free. Radio Paradise sneaked in under the wire. We're on there too. We probably grab another 200 listeners at a time that way.

**Doc** How many simultaneous streams do you run?

**Bill** It tops out at about 800-900. KPIG does about the same. Maybe slightly higher.

**Doc** At what bit rate are most people listening?

**Bill** On Radio Paradise, about 600 out of 800 will be listening at 128k.

**Doc** Due mostly to cable and DSL, I would guess.

**Bill** Yes. 128k streams work just great on cable and DSL.

**Doc** Isn't there a fundamental inefficiency involved as that number goes up?

**Bill** Yes, it's incredibly inefficient, but I don't think that's going to change. And we're at the point where bandwidth and hardware are so cheap, or headed that way, that there just isn't that much of a return on reinventing the whole infrastructure of the Internet to make it more efficient.

**Doc** Are you worried about what Disney and RIAA and these other creeps are trying to do with legislation right now?

**Bill** Yeah. If they get their way, they will drive people like me out of business. There would be no way to cost-effectively do what I want to do here.

**Doc** It would be endless digital rights management everywhere.

**Bill** I can't believe that's going to work—that they are going to shoehorn that thing through. There is just too much standing up against it. The biggest is reluctance on the part of consumers. They've had too much of a taste of how it oughta be, with the free exchange of MP3 files, for them to trade that for anything else.

**Doc** Yet when Napster died, not that many people cried. They just kind of moved on.

**Bill** Well, there are any number of alternatives. There's Gnutella, which has a good Linux client. Mac too. True, none of them are as good as Napster was at its peak.

**Doc** So you're not worried.

**Bill** No, I'm really optimistic.

email: doc@searls.com

**Doc Searls** is senior editor of *Linux Journal* and coauthor of *The Cluetrain Manifesto*.

Archive Index Issue Table of Contents

Advanced search

# Dealing with Patents in Software Licenses

**Lawrence Rosen**

Issue #93, January 2002

Advice on the ways that patents affect software licenses and the kinds of retaliation clauses you should consider.

Many members of the Open Source community oppose software patents. Software patents, they say, hinder the advancement of software art, and as such, counter the beneficial effects of open and published source code. The risk of infringing a copyright is much less than the risk of infringing a patent. You can avoid infringing a copyright by employing good clean-room practices and writing your own independent version of copyrighted software. On the other hand, a software patent can stop you from making, using or selling the patented invention, even if you didn't copy the inventor's software. This means that you may not be able to avoid infringing a patent no matter how careful you are. Someone you never even heard of can inform you that he or she has a patent and, if you cannot invent around that patented invention, your open-source project may be stopped dead in its tracks.

Like them or not, however, software patents are a reality. Software patents have been blessed repeatedly by Congress and the courts, and by the laws of many other countries. Given this reality, it is important to understand the implications of patents for software licenses so that you can select a license that meets your philosophy and goals.

Consider, from the viewpoint of a licensee of open-source software, three kinds of patents: 1) patents owned by the software licensor, 2) patents owned by third parties and 3) patents owned by the licensee or by the licensee's downstream sublicensees.

Licensor patents: suppose you took a license for some open-source software and then discovered that the licensor has a patent on that software that was not included in the license grant. Without a license to that patent, you could not make, use or sell the software. Whenever I review a software license for a

licensee, I make sure there is an express grant “under claims of patents now or hereafter owned or controlled by licensor, to make, use, sell, offer for sale, have made, and/or otherwise dispose of licensed software or portions thereof.” Many open-source licenses, including the BSD license, contain no such provision. In those cases, a patent license may be implied, but I don't recommend relying on an implied license. Wherever possible, make sure you have an explicit license to any necessary patents held by the licensor.

Third-party patents: a software licensor may not be aware of all patents that apply to its software. Some third party suddenly may announce that it owns a patent that covers some aspects of the software. As a licensee of infringing software, you may have to stop using the software despite the license. To deal with this situation, proprietary-software licenses often include an indemnity clause, by which the software licensor indemnifies its licensees against third-party patent claims, promising to refund license fees, provide non-infringing versions of the software or obtain licenses to third-party patents, if third-party patents come to light. But open-source licenses usually don't contain indemnity clauses because licensors of open-source software usually do not collect license fees sufficient to cover the potential costs of the indemnification. So, for most open-source software, the license is “as is” without any warranty of non-infringement. Open-source licensees beware! The risk of third-party patents is usually borne by the licensee.

Licensee patents: the issue of licensee patents is much more subtle than with licensor and third-party patents. Licensors often include so-called “patent retaliation” clauses in their licenses to prevent licensees from using patents offensively against the licensor. Because this issue evokes strong feelings that justify careful discussion, I'm saving that topic for its own column next month.

Whatever your philosophy about software patents, it is important to understand the ways that patents can affect software licenses. It is not enough just to say “I don't like software patents.” Whether you out-license your software to others, or you in-license other's software for your own use, you should make sure that the license fairly expresses your own philosophy and goals relating to patents.

Legal advice must be provided in the course of an attorney-client relationship specifically with reference to all the facts of a particular situation and the law of your jurisdiction. Even though an attorney wrote this article, the information in this article must not be relied upon as a substitute for obtaining specific legal advice from a licensed attorney.

email: lrosen@rosenlaw.com

**Lawrence Rosen** is an attorney in private practice in Redwood City, California (www.rosenlaw.com). He is also executive director and general counsel for Open Source Initiative, which manages and promotes the Open Source Definition (www.opensource.org).

Archive Index Issue Table of Contents

Advanced search

# Coyote Point Equalizer

**Logan G. Harbaugh**

Issue #93, January 2002

Load balancers are devices that distribute client requests from the Internet to a virtual cluster of servers (often called web farms).

Given the popularity of Linux with many ISPs, it behooves the Linux system administrator to be aware of load balancing since most ISPs use load balancers to add scalability and fault tolerance to the web servers they provide to their customers. Load balancers are devices that distribute client requests from the Internet to a virtual cluster of servers (often called web farms). With a virtual cluster, more requests can be handled than a single server could process, and any server in a cluster can fail without interrupting service because the load balancer will simply bypass the disabled server, and the other servers in the cluster will continue to operate.

A load balancer creates a virtual IP address. For example, if the address resolved by DNS for www.foo.com is 192.72.166.240, that address actually is the load balancer. Therefore, any traffic sent to www.foo.com actually is directed to the load balancer, which then directs requests to one of the servers in the web farm. In a typical scenario, the load balancer is connected to two networks. One Ethernet port is given the IP address of the web site, and the other port is connected to the network where the actual servers are connected.

In the same sense that multiple web servers can reside on a single physical server, load balancers can create many virtual clusters on the same group of servers, each with a different virtual IP address but directing requests to the same servers. This enables an ISP to replicate content to as many servers as necessary in a web farm, so that one URL might be spread across ten servers, while another URL served by the same load balancer would only reside on three of the ten servers.

Load balancers use different algorithms to distribute loads among the servers in a web farm. The earliest versions used a round-robin method, simply

rotating through the list of servers, sending each successive request to the next server on the list. The problem that quickly became apparent was that different requests could produce vastly different loads on the server—running a CGI script used much more of the server's processing power than downloading a graphic, although the graphic may use a great deal more network bandwidth. Newer algorithms try to address this by sending the next request to the server that is responding the fastest or that has the least number of users connected to it.

There are several types of load balancers: switches, software-only products and appliances. Load-balancing switches are physically the same as the usual 10/100/1000 Ethernet switch, but with load-balancing functionality added. Software-only products require a PC with two Ethernet interfaces and usually take considerable expertise to set up. Appliances generally are based on a rackmount, Intel-based PC running UNIX, usually FreeBSD, preconfigured to run a load-balancing application.

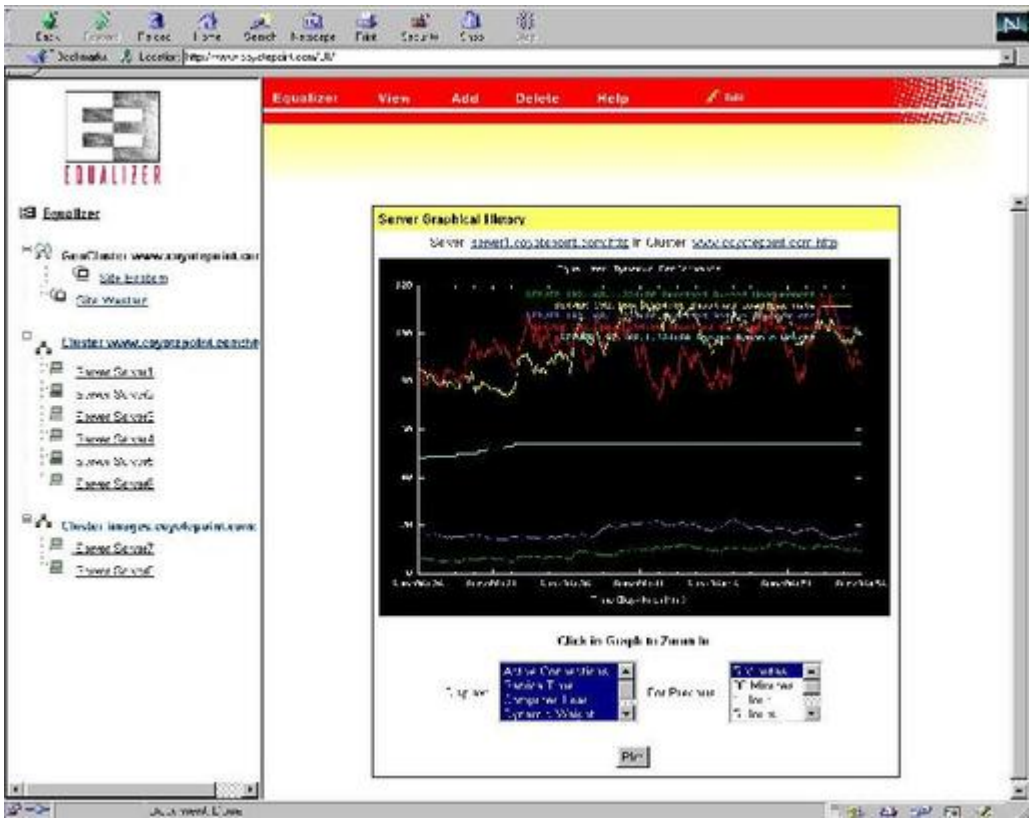
The Coyote Point Equalizer E350 is a good example of the load-balancing appliance: a 2U (3.5") rackmount industrial PC chassis, with a Pentium III processor, 64MB RAM and two 10/100 Ethernet interfaces. It features a removable hard drive, which would allow for upgrades or repairs without having to swap the entire unit. By the time this article is published, the E350 will have been changed from a 2U chassis to a 1U (1.75") chassis.

The E350 can be administered through a Telnet session with ssh or via a web browser. The browser must support JavaScript. Before the device can be administered, an initial setup has to be completed in order to give the E350 a hostname, IP addresses, subnet masks for the two Ethernet interfaces and the default router for the external interface, plus the IP address of the DNS server, time and date, and the password for the administrator interface.

The Equalizer can be preconfigured by Coyote Point at no cost; the customer fills out a one-page form, and the Equalizer arrives with all the basics set up, eliminating the need for the initial setup via serial terminal. Since this one was not ordered with the configuration preset, I set up the serial connection and entered the basic configuration information, then logged in to the box from a browser to set up the virtual cluster. Both the initial configuration and the setup of the cluster went smoothly.

The management interface is straightforward and clear. While some network devices seem to have been designed to be administered only by a command-line interface, with the browser interface an afterthought, the Equalizer's browser interface is a strong application that clearly has been designed to be easy to use. It features strong reporting tools that provide a graphic display of

loads on both the cluster and individual servers. It allows historical analysis, so the administrator can see trends and take action before loads become too high. The administrator can set up triggers that will run a script or send an e-mail, alerting the administrator if a site or server fails, for instance.




Screenshot of Equalizer's Server Graphical History Chart

The documentation is clearly written, in a single printed manual. This is a special advantage in Linux shops, since some competing products provide documentation only on CD and in PDF format, which can be problematic to read.

I set up a single cluster with three servers. The E350 allows the administrator to choose from a number of different load-balancing algorithms: fastest server response time, least number of requests, static weighted assigned values, round-robin or actual server load measured with optional server agents. Coyote Point offers sample C code for writing server agents but does not include agents.

I tried all of the algorithms, using RadView Software's WebLoad to generate traffic against the virtual cluster. All of the algorithms worked, providing good distribution of loads between the servers. The Equalizer also was able to detect a failed server immediately, based on either a ping failure or the failure of the web server to return a proper response to content verification. The Equalizer allows you to check a specific URL and verify the return string to ensure that content is available on a server, rather than just relying on the server

responding to a ping or TCP port check, which could return a value even though the web server had hung up.

The E350 also is available with a geographic load-balancing option, which I did not test. This $2,995 option allows load balancing across multiple sites so that a user can be directed to the site that will provide the best performance (not necessarily the closest site physically).

Once a user has been directed to a particular server, it is sometimes desirable to ensure that they stay with that server throughout a session. For instance, during an e-commerce session, the user should remain connected to the same server, since another server would not have the information on the shopping cart for that user. Normally each request during a session is directed to the least-loaded server, which could change during a session. The way around this is to make sessions “sticky” by identifying the user in some way so that all requests from that client can be sent to a single server.

In this area, the Equalizer is somewhat less sophisticated than other devices: it provides sticky sessions based only on IP source address, and it doesn't support cookie, URL or secure socket layer (SSL) session ID-based persistence. With large numbers of users coming through AOL, which may change a user's IP address multiple times during a session as well as the growing use of network address translation (NAT) in most corporate internet gateways, this might be an issue for some users.

Tech support includes a business-hours, toll-free phone and e-mail support, and optional 24/7 phone and e-mail support with on-site hardware repair.

The Equalizer is available in three models: the E250, E350 and E450, as well as redundant versions that include two controllers with failover capability. The models vary in the number of servers and clusters they support: the E250 supports 64 virtual clusters of up to eight servers each, up to 64,000 simultaneous connections and is targeted at sites with T-1 access; the E350 supports an unlimited number of 16-server clusters, up to two million simultaneous connections and is targeted at sites with T-3 connections; and the E450 supports an unlimited number of 64-server clusters and up to four million simultaneous connections and is targeted at sites with up to 100Mbps connections.

While most load balancers are used to create web farms, they also can be used to scale or provide redundancy for other kinds of servers. The Equalizer supports UDP load balancing, which supports UDP protocols such as DNS, Radius and WAP, as well as network-attached storage devices.

Product Information/The Good/The Bad

**Logan G. Harbaugh** (lharba@awwwsome.com) is a freelance writer specializing in networking. He has worked as an information technology manger and manager of systems integration and has been a networking consultant for more than 15 years. He has also written two books on networking.

Archive Index Issue Table of Contents

Advanced search

# Letters

**Various**

Issue #93, January 2002

Readers sound off.

The November 2001 issue of *LJ* arrived in the mail yesterday, and I was surprised to see on page 56 a photo of a bicycle! But after thinking about it, I thought, if any bicycle is appropriate for a Linux magazine, the Bike Friday is it. It is a quality product, the support is superlative, and it can do so many things that outsiders don't expect. And yes, I'm talking about the bicycle—and Linux. See, a perfect match! Incidentally, one of those things that the Bike Friday bicycles can do is fold. It's not mentioned in the article, but one of their biggest attractions is transporting them in airline-sized suitcases, which then can be towed behind the bicycle as a trailer.

—Larry Varney

## Palm Comments

I just received your November 2001 issue, and I'm glad to see an article covering the topic of using a Palm device to talk to one's Linux box. This is a useful project that helps to prove that Linux is more than ready for the desktop, yet has not gotten the attention it deserves.

However, there are some problems with Coppieters' and Velghe's otherwise useful article. First was their use of pilot-link 0.9.3, which is an old release, and referring users to the old FTP site at Ryerson University. Pilot-link is currently at release level 0.9.5, and the source code for this can be found at www.pilot-link.org. I would also like to take the time to single out David A. Desrosier, who picked up this project two years ago and has kept it alive. Currently he is helping to add support for USB to pilot-link for the next release. And should a reader currently need to use USB to talk to her/his Palm device, there is ColdSync, which can be found at www.ooblick.com/software/coldsync.

I invite your readers who are interested in current developments with Linux and PDAs using the Palm OS to subscribe to the pilot-unix mailing list at pilot-unix@hcirisc.cs.birmington.edu. Besides working on USB support, work has begun to enable the protocol that underlies the PalmPix graphics viewer so that users can load images from their Linux box and view them on their Palm device.

—Geoffrey Burling

### Some Ultimate Advice

In your article "The Ultimate Linux Box 2001: How to Design Your Dream Machine" (unabridged web version, available at /article/5563), you wrote:

> The SB Live! seemed to work with the stock emu10k1.o sound module in Red Hat 7.1, but as it turns out it can't run the earphone-out jack on the LiveDrive.

Actually, it can—it just isn't set up to do so by default. I own one of these cards, a good pair of headphones, and a cheap pair of speakers, so I had reason to look into this. Here's how I got it working. First, download the drivers from opensource.creative.com:

```
cvs -d ':pserver:cvsguest@opensource.creative.com:/usr/local/cvsroot' login
[use the password 'cvsguest']
cvs -d ':pserver:cvsguest@opensource.creative.com:/usr/local/cvsroot' co emu10k1
```

Everything there is under the GPL and changes here get folded into the kernel tree, so there's no real point using this driver over the kernel one. However, there are some utilities included that let you (among other things) enable different inputs/outputs. So compile and install their emu10k1.o if you want, but there's no need to. What we're after is **make tools**. This gives you all sorts of tools for doing fancy things with the card, most of which I don't understand. The only one you need to get the headphones working is emu-dspmgr, located in the utils/mixer directory. With it, you can pipe your choice of inputs to your choice of outputs, e.g.:

```
emu-dspmgr -a'Pcm L:Phones L' emu-dspmgr
-a'Pcm R:Phones R' emu-dspmgr
-a'CD-Spdif L:Phones L'
emu-dspmgr
-a'CD-Spdif R:Phones R'
```

--Andrew Bishop

### Monarch Impostor?

Regarding your latest issue with the front cover depicting the "Ultimate Linux Box", a computer reseller called Monarch Computer at www.monarchcomputer.com is claiming that their product is this box. They've

even gone so far as to modify an image of the *LJ* cover to put their own company's name on the image; the image on their web site reads "*Linux Journal* Ultimate Linux Box Monarch Computer Systems". According to the article, it was another hardware vendor (Los Alamos Computers) that helped put the effort into designing and building this system. So if anybody should get credit by way of business, it should be those guys. This seems very sleazy.

—Mark Travis

*Mark, if you look carefully at the article, you will see that there were two boxes built—one by Los Alamos for Eric Raymond and another by Monarch for LJ* Technical Editor Don Marti. It actually was Monarch's box that appeared on the cover.

—Editor

### No Word Yet

I much appreciated the article by Jan Schaumann, "More than Words" in the November 2001 issue. Not only did I learn some new ways to deal with .doc files, but I was pleased to see him plug TeX and LaTeX. Word processors were invented for those who want to aid and abet in their own victimization, and the users of word processors deserve all the problems and version incompatibilities they experience.

I've been a TeX and LaTeX user since 1985, and everything I've written since then still runs through typesetting with the same results. Using a simple text editor, one can generate PostScript files, Portable Document Format files and HTML files—that is, one source document can be a printable manual, a downloadable manual and a web presentation. We even use LaTeX to generate e-commerce click-to-buy pages, the generation of which is easily automated. The make utility automates generation of all of the mentioned outputs with dependency checking. Multiple authors can take part in an "expression" and have it present a consistent and always up-to-date content. Examples of our usage can be found at www.amplepower.com and www.pwrtap.com. Word processors? No thanks, I have work to do.

—David Smead

### Erratum: "2001 Readers' Choice Awards"

(*LJ*, November 2001)

The article read:

> Favorite Desktop Environment
>
> 1. KDE 2. GNOME 3. Window Maker
>
> This was one of the most popular categories, and KDE is the clear winner, receiving 40% of all votes. GNOME came in second with 24.5%, and the favorite write-in was xfcr. And special mention, of course, for the command line.

The “favorite write-in” mentioned above should be spelled XFce and not xfcr.

—Chuck Meadwww.xfce.org

Archive Index Issue Table of Contents

Advanced search

# UpFront

**Various**

Issue #93, January 2002

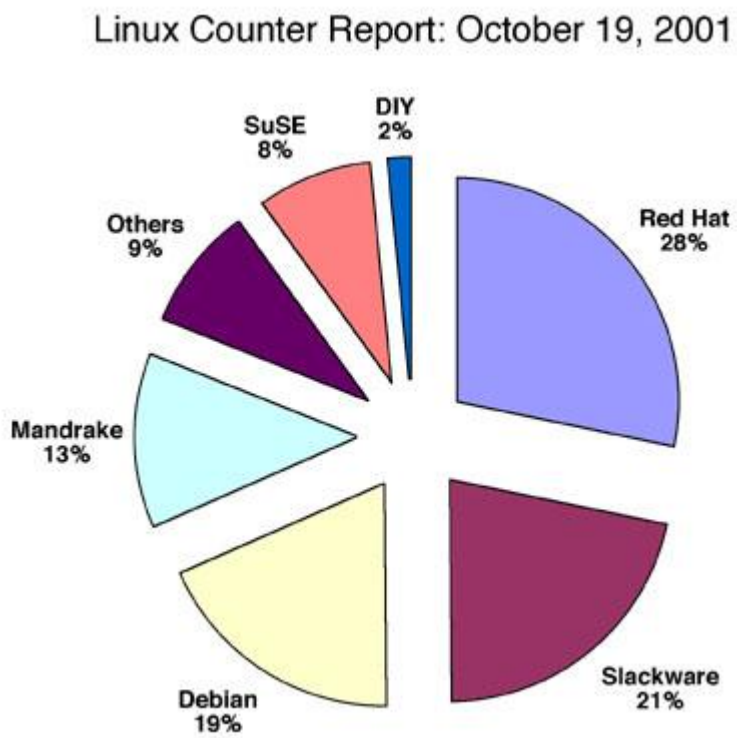
Stop the Presses, *LJ* Index and more.

## Distro Shares: the Voluntary View

Linux Counter (counter.li.org) is a voluntary effort. It only counts people who bother to register (and their machines). As of October 19, 2001, Linux Counter had attracted close to 200,000 individual registrations from nearly 200 countries.

Here in Upfront we try to show various Linux statistics: from Netcraft, Tucows, Evans Data and other sources. This month we thought it would be a fun idea to see what users themselves are saying about their distribution choices, including DIY: do it yourself.

The pie chart shows the results. If you don't like 'em, go vote with your own registration at the Linux Counter site.




Linux Counter Report: October 19, 2001

—Doc Searls

### *LJ* Index—January 2002

1. Millions of people who watched streamed media of the September 11, 2001 terrorist events: 21
2. Billions of dollars in federal relief requested by New York City: 54
3. Gross Domestic Product (GDP) in billions of dollars of Peru: 54
4. Thousands of Korean Air pilots and flight attendants checking schedules using IBM's Linux-powered eServer: 3
5. Billions of e-mail accounts anticipated by the end of 2001: 1
6. Years it took TV to reach 1 billion viewers: 50
7. Years it took telephony to reach 1 billion users: 100
8. Position of Germany among countries with highest penetration of Netscape browsers: 1
9. Netscape browser share percentage in Germany: 20.26
10. Netscape browser share percentage in the US: 15.79
11. Netscape browser share in the world: 13.17
12. Number of countries with users registered to Linux Counter: 188
13. Number of persons registered with Linux Counter as of October 19, 2001: 195,900

14. Number of machines registered with Linux Counter as of October 19, 2001: 111,942
15. Low end of Linux Counter's estimated number of Linux users: 3,918,000
16. High end of Linux Counter's estimated number of Linux users: 97,950,000

### Sources

1: *USA Today*, quoting Nielsen/NetRatings

2-3: *Time Magazine*

4: IBM

5-7: Strategic Policy Research, Inc.

8-11: StatMarket (www.statmarket.com)

12-16: Linux Counter (counter.li.org)

### Stop the Presses: Mixed Happenings on the Legal Front

Two unrelated legal developments are taking shape as we go to press—one DVD-related and one involving Microsoft. In the first, a California appellate court unanimously shot down a trial court injunction that banned publication of DeCSS code that decrypts DVDs so they can be played on any computer, and not just on players manufactured by members of consumer electronics cartels. This case is actually one of two DVD-related cases. The other, Universal Studios and United States of America vs. Corley, is now in the US 2nd Circuit Court of Appeals. At the original trial in New York City, the publisher of *2600 Magazine* lost the right to publish source code and information about DVDs, in front of a judge who used to work for Time Warner. In the Microsoft case, the US and Microsoft came to a preliminary agreement that likely will bring their longstanding legal battle to an end.

At issue in the DeCSS case were the first ammendment rights of those publishing the code vs. the trade secret rights of entertainment companies distributing the DVDs. Those companies objected specifically to the publication on the Web of DeCSS software written by programmers in the fall of 1999 as part of an effort to create a DVD player for computers running Linux. In early 2000, the DVDCCA, the movie studios' DVD licensing organization, filed a lawsuit against hundreds of programmers and web publishers seeking to ban DeCSS publication. Santa Clara County Judge William Elfving granted the injunction request on January 21, 2000. The appellate court ruled that Elfving violated the First Ammendment rights of defendant Andrew Brunner by ordering him to remove the code from his web site. The lower court based its

rulings on trade secret misappropriation, even though Brunner had found the program in the public domain (on Slashdot) and simply republished it. The case is still expected to go to trial next year before Judge Elfving.

In the matter of the US vs. Microsoft, most pundits agreed that the settlement favored Microsoft, which the original judge on the case, Thomas Penfield Jackson, had called for breaking in two. Dan Gillmor of the *San Jose Mercury News* wrote:

> This deal, assuming it takes hold, is a love letter to the most arrogant and unrepentant monopolist since Standard Oil. It's an invitation to keep on plundering and whacking competition in the most important marketplace of our times, the information marketplace.

John Borland of CNET called the settlement “a reward, not a remedy”. Dave Winer of Userland, an independent commercial developer that worked with Microsoft on the SOAP and XML-RPC protocols, wrote:

> It clearly doesn't go far enough and requires the government to be involved in the architecture of Microsoft's operating system and network services, such as SOAP, in an intimate and impractical way.

While the settlement would leave the company intact, it would end Microsoft's practice of forcing its hardware OEMs to distribute Windows. It also would force the company to share elements of its operating systems, giving competitors a little more room to actually compete with Microsoft's own applications. Winer's “intimate and impractical” remarks refer to a “Technical Committee” that will be co-appointed by Microsoft and the Court and installed at Microsoft to oversee compliance.

The one apparent upside for Linux is the release of hardware makers from Microsoft OEM contracts that require distribution of Windows. At least conceptually this leaves a much more open hardware channel for Linux distribution.

—Doc Searls

### They Said It

No matter what they do, companies are sure to annoy someone.

—Deborah Branscum

Microsoft is after all of our entertainment technologies. It is an assault on so many fronts that it is easy to lose track.

—David Strom

There is no law that a company has to stay in business. On the contrary, there is a law that everything man creates is mortal. It is rare for a company to be successful for more than 25 years. The idea that companies are immortal is a Wall Street misunderstanding. The main impact of the Internet is not economic; it is psychological. There is no new economy. The Internet greatly extends the old economy, okay?

—Peter Drucker

No financial man will ever understand business because financial people think a company makes money. A company makes shoes, and no financial man understands that. They think money is real. Shoes are real. Money is an end result.

—Peter Drucker

Advertising is the last resort of the incompetent.

—Protobot

Although it is true that writing open-source software is not on the same level as running into a collapsing building to save lives, it is true that electing to write open-source code is by no means a subversive activity. It is, in fact, a little slice of selflessness—something that America applauds.

—Russell Pavlicek

Three pearls of wisdom from e-failures: 1) Free does not equal profit. 2) Start small and then get big. 3) Geeks don't know crap about business.

—Michael Jardeen

Leaders are visionaries with a poorly developed sense of fear and no concept of the odds against them.

—Dr. Robert Jarvik

If there are 1,000 people who hear you, only 500 will take the time to listen, 300 will understand what you said, 100 will do something about it and 50 of those will be negative. We can hope that the other 50 will do something positive.

—John “maddog” Hall

Today, ROI calculations are all important, and open-source is the savior for enabling sophisticated applications to be built at a price point that allows a reasonable return on investment.

—David A. E. Wall from Yozuns white paper

Archive Index Issue Table of Contents

Advanced search

# High Seas Adventure

**Richard Vernon**

Issue #93, January 2002

A cruise to the land of lunacy, or Survivor geek style?

The Caribbean's a great place to get geeky if you can keep the sand out of your laptop. Of the many cruises organized by Geek Cruises, this was the first one devoted to Linux. *Linux Journal* had the opportunity of cosponsorship, and it's something we look forward to repeating next year.

Besides being treated to intimate sessions and abundant chances for one-on-ones with some of the most prominent names in the Linux community, for us the cruise was a chance to get to know many of our readers (and even some of their families).

The geek sessions were balanced very well with time off in some fabulous ports of call—well, except Puerto Rico. But even there, Geek Cruise organizer, “Captain” Neil Bauman, arranged a special excursion for geek cruisers to see the home of the world's largest single-dish radio telescope, the Aricebo Observatory, where much of the movie *Contact* was filmed, and where scientists from around the world study things out of this world 24 hours a day, 365 days per year.

Of course we weren't so lucky in choosing the lunch restaurant that day. It was a buffet that ran out of food after half of us had made it through the line. Fortunately, the entertainment value of seeing Richard Stallman spill his soda (accidentally of course) on the world's largest baby who was sleeping in his playpen near the cash register was some compensation for my hunger. I couldn't help but laugh internally—the irony was too much, but I actually felt more sorry for Richard than the baby; he felt so badly about it. In some cosmic form of retribution my own two-year-old daughter spilled her soda (same flavor as Richard's) all over my lap, leaving me looking as if the excitement of the observatory was too much.

In contrast to Puerto Rico, the other ports of call were absolute representations of paradise. At our last stop, Holland-America's private island, Half Moon Cay, I rented a Sunfish sailboat with my eight-year-old daughter, Geneviéve. We sailed to the end of the bay and still could see straight to the ocean floor. For someone used to sailing the dark water of Washington's Puget Sound, it was a unique experience.

Next year's Linux Lunacy cruise is rumored to be planned for the Pacific, down Mexico way. Bring your sense of humor, your goodwill and join us!

**Richard Vernon** is editor in chief of *Linux Journal*.

Archive Index Issue Table of Contents

Advanced search

# Best of Technical Support

**Various**

Issue #93, January 2002

Our experts answer your technical questions.

I share two boxes, one Windows and one Linux, with one KVM. Both work when I start up the desktop, in this case GNOME/Enlightenment. When I switch from the Mandrake desktop to the other system, then back again, I lose mouse support entirely. I've checked cables, restarted the gpm dæmon and pressed Ctrl-Alt-Backspace to leave the desktop. When I restart the desktop, the mouse is detected again, but only for as long as I don't switch to Windows and then back again.

With the Windows side, I can switch to Linux and back with no problem, and every time I start the desktop on Linux it works—but only until I switch the screen away and back again. Any ideas as to what this needs to fix? Linux/Mandrake 7.1 is running on a Dell P90 (old) with PS/2 mouse, gpm runs with **gpm -t ps/2**. Could some other dæmon I'm not running because of security be what's causing the problem? I have amd, atd, innd, lpd and portmap disabled.

—Dave Dennis, dmd@speakeasy.org

This is most likely not a problem with the Linux setup. PS/2 mice have a configuration that is initialized during startup. A KVM is responsible for restoring this configuration on switch back to a machine; Linux is totally unaware of the switch.

—Christopher Wingert, cwingert@qualcomm.com

## Didn't Run LILO

I recently upgraded my kernel and forgot to run LILO before rebooting. Now, if I start up from a different disk, that disk can be mounted and fsck shows it is clear. However, I can't boot from it.

—Willie Strickland, willie@istrick.com

Use the boot disk to boot from the hard drive with a command similar to this at the LILO prompt:

```
linux root=/dev/hda1
```

Then edit your lilo.conf and run LILO, as if you'd just installed the new kernel.

—Ben Ford, ben@kalifornia.com

### I Have No “mail” and I Must Mail

I have networked Linux machines on my home network consisting of two desktop machines and a laptop. I would like to get my mail on any machine but can do so only on the older desktop machine. I have set up the Netscape preferences identically on all machines. On the newer machine and the laptop I get the message “Netscape unable to locate the server mail”, when I try to retrieve mail. The server “mail” is the name given by my ISP (Cox@home), which works perfectly fine on the older machine. On the other machines, when I try to get new messages, Netscape always asks for my password even though in the preferences I have explicitly selected the “remember password” button, so I'm wondering if NS is reading the wrong preferences file.

—Eric Smith, esmith289@home.com

Sounds like your one working machine has the Fully Qualified Domain Name (FQDN) for your ISP and the others do not. Try adding the rest of the hostname to the configuration for Netscape (i.e., mail.example.com, if example.com is your domain name). Alternately, you could update your /etc/resolv.conf “search” configuration line and add the correct domain name so that you don't have to type in all the time.

—Christopher Wingert, cwingert@qualcomm.com

To check that you have edited /etc/resolv.conf correctly, do a **host mail** from the shell to see what “mail” is resolving to.

—Don Marti, info@linuxjournal.com

### audoupdate Can't Find Directory

I am trying to use autoupdate 3.1.5. When I type **autoupdate**, I get this error message:

```
CWD failed no such directory or file
```

When I run **autoupdate --debug 2**, it is able to log in as anonymous user, then it says:

```
CWD failed.
Error: Failed to check directory at ftp.redhat.com:
pub/redhat/linux/updates/7.1/en/os
no such file or directory.
```

I tried typing **ftp ftp.redhat.com**. I am able to change the directories to /pub/redhat/linux/updates/7.1/en/os as anonymous user.

—Adharsh Praveen R., adarsh@multitech.co.in

You should add a "/" in front of the directory passwd to autoupdate, i.e., **/pub/redhat/linux/updates/7.1/en/os**.

—Christopher Wingert, cwingert@qualcomm.com

### Where's /dev/sdc?

I have two IDE devices, a 40GB WD HDD and a CD-RW drive. I also have a SIIG SCSI card. I believe that Linux is recognizing my SCSI card, but for some reason I cannot get access to my SCSI drives. When I try **/dev/MAKEDEV sdc[0,1, ..., n]**, it tells me:

```
don't know how to make sdc[n].
```

My SCSI hdd is set to id 3. My SCSI CD-ROM is set to id 5 and my SCSI card is defaulted to id 7. Everything is terminated properly. What's the deal?

—Derrick Blackwell, db101055@hotmail.com

You seem to expect the HD with SCSI address 3 to be /dev/sdc. Linux doesn't work like that: the lowest-addressed SCSI HD is always /dev/sda, regardless of its SCSI address. The next-higher-addressed SCSI HD is always /dev/sdb, and so on. Your SCSI CD will be /dev/scd0 or /dev/sr0—both names work equally well.

—Scott Maxwell, maxwell@ScottMaxwell.org

### What the fsck? The Superblock Could Not Be Read

When I **e2fsck /dev/hda**, I get the following message:

```
The superblock could not be read or does not describe
a correct ext2 filesystem. If the device is valid
and it really contains an ext2 filesystem (and not
swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an
alternate superblock: e2fsck -b 8193 <device>
```

When I do as it says (<device> = dev/hda), I get the same message back. This is is the fourth time it has happened. The other times I just reloaded Red Hat, as I had not lost anything significant. This time I prefer to not lose programs I had working.

—Bob Wooden, bwooden@computelnet.com

/dev/hda is a drive. While it is possible to install a filesystem on a drive, this is most likely not the way it was installed. You should do a **fdisk -l /dev/hda**, which will show you all the partitions installed on the hda drive. Most likely, your partition is /dev/hda1 unless you are dual booting.

—Christopher Wingert, cwingert@qualcomm.com

### /proc/kcore: 0wn3d!

I have two servers running Red Hat 7.1. I run Tripwire on both servers. Tripwire has twice reported that the file /proc/kcore has changed. Is this something that happens? No reboots between the occasions.

The file /lib/libc-2.2.2.so did change checksum, but nothing else was changed; dates, inode, etc., were the same. **rpm -V glibc** also revealed that the file MD5 checksum was altered. Since I am running a public web server on this machine, I reinstalled the glibc package. Then Tripwire complained about the dates and inodes; this is natural when you have reinstalled a package. I believe I have a well-setup firewall. How can the checksum of libc change? Why does the checksum of kcore change?

—Magnus Sundberg, Magnus.Sundberg@dican.se

I don't know why libc changed, but I know why /proc/kcore changed: /proc/kcore is a pseudo-file representing the system's physical memory. It's only natural that the contents of physical memory would vary with time. If it didn't, your system would be a lot less useful. So don't worry about that one.

—Scott Maxwell, maxwell@ScottMaxwell.org

### If I Could Walk That Way, I Wouldn't Need Aftershave!

I am having trouble using two NICs in my Red Hat 7.2 (2.4.3-12) machines. Both NICs are recognized, and I can configure them. But I have two T1s from different providers, thus on different IP networks. I want to multihome these machines, but I cannot figure out how to add a default route for each NIC.

—Mike Kercher, mike@CamaroSS.net

You cannot add two default routes. You have to tell Linux which traffic should go to which network. Sounds like what you are trying to do is load balancing on the two T1s (or possibly redundancy). Check out the EQL or Bonding driver. You also could route via BGP. Check out the Advanced Routing HOWTO at www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html.

—Christopher Wingert, cwingert@qualcomm.com

### Yes, Sir, We Have Aftershave—Walk This Way Please

I have about 300 Linux servers and the floor space in the data center is at a premium. I need to access the servers as console for administration purposes, but they do not have monitors. Is there some way I could get a serial terminal connected as a console?

—Karthik, nkk@hotmail.com

Check out the Serial Console HOWTO at www.linuxdoc.org/HOWTO/Remote-Serial-Console-HOWTO.

—Christopher Wingert, cwingert@qualcomm.com

### Backup Chokes

I use a DDS4 tape drive on a Compaq Alpha (Red Hat 7.1) to back up several filesystems on other Linux workstations (PCs with Red Hat 7.0 and 7.1). I've set up ssh so DSA authentication allows me to run a simple backup script at night via cron, without being prompted for a password. I use tar (gtar version 1.13.17) and dd (GNU fileutils 4.0x) to store the files on tape. Somewhere down the tar/dd combination I get the message:

```
select: Bad file descriptor
```

The backup script then stops tar/dd-ing and proceeds normally with the rest of the instructions. What am I missing?

—Martin Olivera, molivera@ucsd.edu

This error has shown up as a result of a bug in early versions of OpenSSH. The first thing to try is to upgrade ssh on all the systems to the latest stable version.

—Don Marti, info@linuxjournal.com

Archive Index Issue Table of Contents

Advanced search

# New Products

**Heather Mead**

Issue #93, January 2002

Cyclades, webMathematica, YDL 2.1 and more.

## Cyclades TS400, TS800, TS3000

Cyclades Corporation has released the TS400, TS800 and TS3000, the newest members of the TS-Series, which are console servers used for out-of-band management of console ports in server farms and clusters. Featuring the same software as the other TS servers, the TS400, TS800 and TS3000 feature 4, 8 and 48 ports, respectively, in a 1U rack space. The TS3000 combines SSH, IP filtering and RADIUS with off-line data buffering and break-safe operation. Each of the TS-Series can also be used as a PPP server to provide access to dial-up clients or telecommuters using analog modems.

Contact: Cyclades Corporation, 41829 Albrae Street, Fremont, California 94538, sales@cyclades.com, www.cyclades.com.

## webMathematica

webMathematica is software that allows users to add interactive calculations to the Web. With webMathematica, users can build web sites that provide specialized calculations for fields such as electrical engineering, compute and visualize data with a browser, deliver courseware and interactive books, and provide active functionality for technical documentation. Built on Java servlets, webMathematica is compatible with any web server, servlet engine or application server that supports Servlet 2.0 API or higher. It currently is available under a Professional or Amateur license for Intel-based Linux platforms, with other platforms to follow.

Contact: Wolfram Research, Inc., 100 Trade Center Drive, Champaign, Illinois 61820, www.wolfram.com.

## YDL 2.1

Terra Soft Solutions announced that Yellow Dog Linux version 2.1 is now available for PPC processors. Updates to this release include making the 2.4 kernel the default and providing NVIDIA GeForce 2 video support. Other new features and updates offered are KDE 2.2.1, XFree86 4.1.0, Mozilla 0.9.3, Ext3 journaling filesystem, support for ATI RADEON video cards and a web-based administration tool. The YDL installation now supports individual package selection and offers Mac-on-Linux 0.9.60, which takes the ROM imaged from the Mac OS partition.

Contact: Terra Soft Solutions, 117 West Second Street, Loveland, Colorado 80537, 970-278-9243, presales@yellowdoglinux.com, www.yellowdoglinux.com.

## Super Mini Optical Mouse

Designed for notebook computers and mobile users, Atek Electronics' two-button Super Mini Optical Mouse is the smallest computer mouse available, measuring 1" × 2.5". It provides the precise cursor control of an optical mouse with the advantage of its small size to be more useful and productive than standard laptop touchpads, trackballs and stick pointers. The Super Mini has a polycarbonate plastic shell and a Kevlar-reinforced three-foot cord. Both the USB and PS/2 versions are compatible with Linux.

Contact: Atek Electronics, 15042 Parkway Loop, Unit C, Tustin, California 92780, 888-889-9990 (toll-free), info@atek.com, www.atek.com.

## Qt 3.0

The newest version of Qt, version 3.0, has been delivered by Trolltech. Qt offers a software development environment to create a single source tree that runs natively on many platforms, now including Mac OS X. New features introduced in version 3.0 are, among others, the ability to build platform- and database-independent database applications; text engine that supports rich text input, editing and rendering; the Qt Designer GUI builder that supports main window development and includes an integrated C++ editor; Qt Linguist for translation of GUIs to different languages; and Qt Assistant for easier searches. Qt v.3 is available under Professional, Free and Educational licenses.

Contact: Trolltech, Inc., 3350 Scott Boulevard, Building 55, Suite 2, Santa Clara, California 95054, info@trolltech.com, www.trolltech.com.

## Linux for Chemistry

The Linux for Chemistry series (Volume 1) is now available from The Random Factory. Linux for Chemistry includes over 1GB of chemistry-related applications all precompiled and compatible with most recent distributions (Red Hat 7.x and SuSE 6.4 or later are recommended). Applications range from molecular visualization and data modeling, to reaction kinematics, spectroscopy and more. A comprehensive on-line documentation library also is provided. A graphical package installation tool is included, as are recent graphics and database releases, security toolkits and remote desktop tools. Linux for Astronomy and Linux for Biotechnology also are available.

Contact: The Random Factory, PO Box 44070, Tucson, Arizona 85733, rfactory@theriver.com, www.randomfactory.com.

Archive Index Issue Table of Contents

Advanced search