Volume 6, Number 11 • November 2000
www.elementkjournals.com/sun
U.S. $11.00

Inside Solaris™

Tips & Techniques for users of Sun Solaris

Making space with partition switching

DOWNLOAD ftp.elementkjournals.com/sun/nov00

by Jerry L.M. Phillips

You knew that it was only a matter of time before *it* happened. You discover the error message "Oct 23 11:41:04 gem unix: NOTICE: alloc: /var: file system full" repeating every few seconds on your database server console screen. Unfortunately, the database server houses the production database for your business. Sales representatives at your company are logged on to the server from 9 to 5 while responding to client requests. No one has complained yet, but you don't have much time.

Temporary measures

In a previous audit of the database server, which you inherited, you noted that the /var partition was probably too small. A flurry of error messages, triggered by a runaway custom daemon that your production database relies upon, has created a huge /var/adm/messages file that has consumed all available disk space on the /var partition, as shown in Listing A.

In order to gain some ground on the daemon errors, you issue the following commands to free up more space on the partition. These commands zero out the contents of the message files:

```
# cp /dev/null /var/adm/messages
# cp /dev/null /var/adm/messages.0
# cp /dev/null /var/adm/messages.1
# cp /dev/null /var/adm/messages.2
# cp /dev/null /var/adm/messages.3
# cp /dev/null /var/adm/utmpx
# cp /dev/null /var/adm/utmp
```

You can also copy the message files to another disk and then zero out the files, if you prefer. Now, restart the custom daemon with the following command:

```
# kill -HUP daemon_pid
```

Thankfully, the errors stop temporarily and allow you some room to breathe. However, you know that this scenario will probably happen again and you might not be around to deal with it when it does. Your best bet is to create a larger /var partition.

Desperate times call for desperate measures

In a resource-rich environment, you may have a spare disk drive that you can partition, build a file system on, mount, and use to create a new /var partition. In this case, you're going to have to use another partition on the system (boot) drive. You determine that the /var partition,

Listing A: *Output from the df command*

```
$ df -k | grep var

/dev/dsk/c0t0d0s3    192799    192799    0    100%    /var
```

In This Issue

element k
JOURNALS
A Division of Element K Press

otherwise identified as /dev/dsk/c0t0d0s3, originally was formatted with 200.39 MB of available space. You also determine that the /export/home partition, /dev/dsk/c0t0d0s7, was formatted with 1.47 GB of available space, and is virtually empty and unused. (Because this is a database server, your DBAs and developers log on to a directory on a different partition.) **Listing B** shows a sample session with the `format` command. You decide to switch the two partitions; /dev/dsk/c0t0d0s7 will become /var and /dev/dsk/c0t0d0s3 will become /export/home.

Listing B: *Output from the format command*

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
/pci@1f,4000/scsi@3/sd@0,0
Specify disk (enter its number): 0

selecting c0t0d0
[disk formatted]
Warning: Current Disk has mounted partitions.
FORMAT MENU:
Disk      - select a disk
Type      - select (define) a disk type
Partition - select (define) a partition table
Current   - describe the current disk
Format    - format and analyze the disk
Repair    - repair a defective sector
Label     - write label to the disk
Analyze   - surface analysis
Defect    - defect list management
Backup    - search for backup labels
Verify    - read and display labels
Save      - save new disk/partition definitions
Inquiry   - show vendor, product and revision
Volname   - set 8-character volume name
!<cmd>    - execute <cmd>, then return
quit
format> verify
Primary label contents:
Volume name = <        >
ascii name  = <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
pcyl  = 3882
ncyl  = 3880
acyl  =    2
nhead =   16
nsect =  135
Part Tag        Flag Cylinders   Size      Blocks
0    root       wm   0 -  379    400.78MB  (380/0/0)  820800
1    usr        wm   380 -  759  400.78MB  (380/0/0)  820800
2    backup     wm   0 - 3879    4.00GB    (3880/0/0) 8380800
3    var        wm   760 -  949  200.39MB  (190/0/0)  410400
4    swap       wu   950 - 1424  500.98MB  (475/0/0)  1026000
5    unassigned wm   1425 - 1804 400.78MB  (380/0/0)  820800
6    usr        wm   1805 - 2456 687.66MB  (652/0/0)  1408320
7    home       wm   2457 - 3879 1.47GB    (1423/0/0) 3073680
format> quit
```

Working with partitions on the system (boot) drive is a special situation, so you wait until all of the database users log off and shut down the system. The /var partition needs to be mounted during multiuser mode and multiuser network mode, because Solaris regularly has to update various log files, and so on, that reside in the /var partition. Thus, it's best that you switch the two partitions while Solaris is in single-user mode:

```
# shutdown -y -i 0 -g 0
```

After the system is shut down, you boot into single-user mode:

```
Ok> boot -s
```

Once the system boots up, you change to the /var directory and issue the following commands that relocate the contents of /var to /export /home. Be forewarned, this may take a while:

```
# cd /var
# tar cf - * | ( cd /export/home; tar xpf - )
# rm -rf /var
```

The first command, `cd /var`, changes to the primary directory that you wish to relocate. The second command,

```
tar cf - * | ( cd /export/home; tar xpf - )
```

is in two parts. In the first part, you use the `tar` command to archive the contents of the /var directory, including subdirectories within the /var directory, to the file -. The - file means send the contents of the archive to the standard output stream. You pipe the results of the first part of the command, using the pipe (|) symbol, to the second part

```
( cd /export/home; tar xpf - )
```

The second part is in the format of a phrase encapsulated within parentheses. The commands within the parentheses start a new subshell. This construction allows the command following `cd /export/home` access to the standard input stream.

Once you change to the /export/home directory, the `tar xpf -` command extracts the contents of the archive from the standard input stream into the current directory, /export/home, and preserves access rights and ownership of the files in the archive. The third command, `rm -rf /var`, erases the contents of the original /var.

Now you dismount /export/home and dismount /var:

```
# umount /export/home
# umount /var
```

If you receive the error message "`umount: /export/home busy`" or "`umount: /var busy`," it means that you're in a directory on one of the drives that you're attempting to dismount. Simply change to the root directory and reissue the `umount` command(s). Now, you can switch the partitions by changing their mount points:

```
# mount /dev/dsk/c0t0d0s7 /var
# mount /dev/dsk/c0t0d0s3 /export/home
```

(Remember that /dev/dsk/c0t0d0s7 mounted previously as /export/home and that /dev/dsk/c0t0d0s3 mounted previously as /var.) At this point, you can resume the boot process and establish multiuser network mode access by typing the following command:

```
# exit
```

Don't forget to hard-code the changes to the mount points, for /var and /export/home, into your /etc/vfstab file. It's a good idea to make a copy of /etc/vfstab before making any changes to it.

Conclusion

That's all it takes to switch a partition. At last you have a comfortable margin of space for your error logs, etc. Now it's someone else's job to fix that custom database daemon! ✳

DOWNLOAD ftp.elementkjournals.com/sun/nov00

Visualizing CPU activity

by Werner Klauser

You're trying to convince your manager that your system needs more processing power. He tells you he wants some visual information. While vmstat's output would show that your system is rarely getting enough idle time, you know that a listing of numbers won't impress your manager. A graph would be a much more powerful visualization tool. But the thought of using Excel doesn't appeal to you, and regretfully you don't have StarOffice installed on your system. Fortunately, you're a friend of Netscape Navigator and are comfortable with HTML and JavaScript.

In this article, we'll show you how to use a free tool from Netscape for building simple graphs. It's easy to use and allows your graphs to be viewed from just about any browser. By combining the output of vmstat into this tool we will visualize our cpu activity.

Using Netscape's JavaScript Graph Builder

Netscape provides, on an as-is basis, a JavaScript Graph Builder library allowing you to make simple, but useful, bar graphs. While building the HTML code, collect system activity information using the following `vmstat` command:

```
$ vmstat 60 > /tmp/vmstat.out &
```

Begin your vmstat.html file by including the necessary JavaScript Graph Builder library information:

```
<HTML>
<TITLE>System Activity</TITLE>
<HEAD>
<BASE HREF="http://developer.netscape.com/docs/
➥technote/javascript/graph/">
<SCRIPT LANGUAGE="JavaScript1.2" SRC="graph.js">
</SCRIPT>
</HEAD>
```

Note that you can also download the graph.js file and install it locally on your Web server. All the interaction with the JavaScript Graph Builder library takes place inside the body of the HTML document, which lets you place the graph anywhere you want among regular HTML content. Once you decide where in the page you want the graph to appear, add the following lines of code:

```
<SCRIPT LANGUAGE="JavaScript1.2">
  // create a new Graph object sized 800 x 300
  var g = new Graph(800, 300);
```

This calls the Graph object constructor function, which creates a new Graph object. You can set its attributes by adding the following lines:

```
// stack the value on top of each other
g.stacked = true;
// set the graph's title
g.title  = "CPU load";
// set the Y-axis label
g.yLabel = "% load";
// set the scale of the Y-axis to units
// of 10 [%]
g.scale = 10;
// set the X-axis label
g.xLabel = "time";
// set the X-axis to begin at 8:00 am
g.setTime(8, 00, false);
// set the labeling of the X-axis to every 1
// 0 minutes
g.skip = 10;
// set the X data to every 1 minute
g.inc  =  1;
```

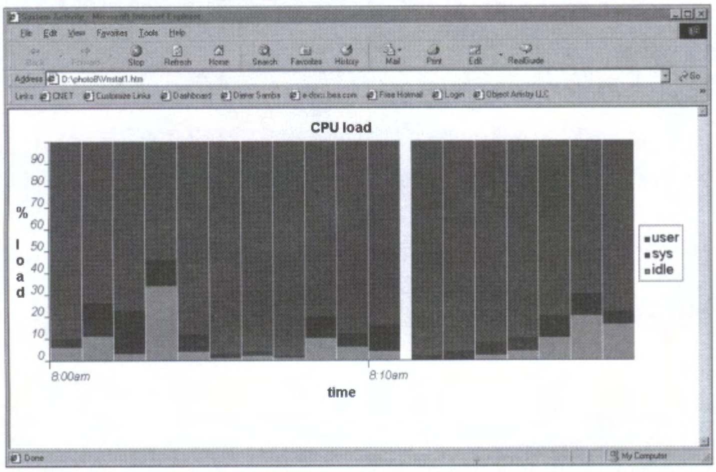
System Activity - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit RealGuide
Address D:\photo8\Vmstat1.htm
Go
Links CNET Customize Links Dashboard Dieter Sambs e-docs.bea.com Free Hotmail Login Object Artistry LLC
CPU load
90
80
70
60
50
40
30
20
10
0
% load
8:00am
8:10am
time
user
sys
idle
Done
My Computer

Figure A: *This is our system activity graphed in Netscape.*

```
// set the graph's legend
g.setLegend("user","sys","idle");
```

Now you need to add the data using vmstat's output, which we have been gathering in /tmp /vmstat.out:

```
$ tail +3 /tmp/vmstat.out | \
   awk 'BEGIN {printf "g.addRow("} \
   {printf "%2d,", $20} END {print " 0);"}' \
   >> vmstat.html
$ tail +3 /tmp/vmstat.out | awk 'BEGIN {printf \
 "g.addRow("} \
   {printf "%2d,", $21} END {print " 0);"}' \
   >> vmstat.html
$ tail +3 /tmp/vmstat.out | awk 'BEGIN {printf
  \"g.addRow("} \
   {printf "%2d,", $22} END {print " 0);"}' \
  >> vmstat.html
```

Now all you need to do is finish your short HTML file with

```
   g.build();
</SCRIPT>
</BODY>
</HTML>
```

Figure A shows the resulting graph with data from our system. This isn't too bad for a few lines of HTML and JavaScript.

Notice that if you hold the mouse over the bar (on Windows/UNIX version of Navigator and Internet Explorer), a small, yellow box appears showing the exact value of the data.

Isn't this enough to convince your manager (and his manager) that you need more CPU power? If want more information on Netscape's JavaScript Graph Builder, take a look at **http://developer.netscape.com/docs/technote/javascript/graph/**. ✳

Determining the number of processors

by Werner Klauser

How do you respond when you're asked how many processors your system has? "Not enough" or "Hmm, I think two" or "I don't know"? Would you like to find out how to determine the number of processors your system has?

As is often the case in the UNIX world, more than one path will get you to the desired destination. Disregarding freeware solutions, we'll show you several methods using Sun Solaris commands. In this case, we'll be using a Sun Enterprise 4500 with two processors. This server supports from one to 14 processors on one to eight system boards.

Print the system configuration

The prtconf command prints the system configuration information. Even the non-verbose output is long. Hidden in its output is the line repeated twice:

```
SUNW,UltraSPARC-II (driver not attached)
```

Every time this repeats, it indicates a processor visible to the system. However, there are easier and more attractive methods to determine the number of processors.

Report per-processor statistics

The mpstat command reports per-processor statistics in tabular form. Each row of the table represents the activity of one processor. The listing from our E4500 is shown in **Listing A**.

Display information about processors

You'll find that the psrinfo command displays information about each of the system's processors. You can execute it without any arguments as follows:

```
# /usr/sbin/psrinfo
0       on-line   since 07/03/00 12:12:47
4       on-line   since 07/05/00 15:19:35
```

Or more interesting:

```
# /usr/sbin/psrinfo -v
Status of processor 0 as of: 07/05/00 15:45:33
  Processor has been on-line since 07/03/00 12:12:47.
  The sparc processor operates at 336 MHz,
        and has a sparc floating point processor.
Status of processor 4 as of: 07/05/00 15:45:33
  Processor has been on-line since 07/05/00 15:19:35.
  The sparc processor operates at 336 MHz,
        and has a sparc floating point processor.
```

Display system diagnostic information

The prtdiag command displays a wealth of system configuration and diagnostic information. You'll have to execute the proper version for your architecture. You can do this easily by using uname:

```
# /usr/platform/`uname -m`/sbin/prtdiag
System Configuration: Sun Microsystems sun4u
➥8-slot Sun Enterprise E4500/E5500
System clock frequency: 84 MHz
Memory size:  512Mb

=====================CPUs =====================

                      Run   Ecache   CPU    CPU
Brd  CPU   Module     MHz     MB    Impl.   Mask
---  ---  -------  -----  ------  ------  ----
 0    0      0      336     4.0   US-II    2.0
 2    4      0      336     4.0   US-II    2.0
```

And even more possibilities...

We'll limit ourselves to these four commands. Don't worry, Sun Solaris allows even more commands such as psradm, psrset, and possibly dmesg. You can also find the cpu information in the console output file /var/adm/messages. ✳

Listing A: *The listing from mpstat for our Sun E4500*

```
# /usr/bin/mpstat
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0  270   4  316   301   86  254   33   30   56    0 12569   13   8  13  66
  4  260   4  328   105   76  256   34   30   54    0 13319   13   8  13  67
```

Windows emulation on Solaris

by Clayton E. Crooks II

There are a variety of ways to run Windows applications on a Solaris machine. Unfortunately, many times the solutions are too difficult for practical use, or maybe even too expensive for many budgets. A software application called SoftWindows 95, available from FWB at **www.fwb.com**, might be the answer for those who are looking for a quick, relatively inexpensive and easy solution.

Emulation overview

Emulating the Intel Pentium processor is a difficult task, but FWB seems to have pulled it off nicely by using compiling and translation advancements. The software is scalable and dependent upon the power and configuration of the hardware on which it executes. Earlier versions of the software (which were acquired from Insignia in October 1999) emulated only an i486 processor.

Windows applications use the Windows libraries extensively to execute standard functions, which are much different than DOS functions. This results in frequent transfers of control and much wider use of code segments. SoftWindows is able to handle this situation without difficulty by sequentially reading the Intel Pentium instructions and translating them into host RISC instructions. The emulator also dynamically identifies frequently executed Intel code segments and compiles them into native RISC code segments. The compiled segments are then cached and run directly whenever the Intel segments are executed, resulting in much faster and more stable code execution.

Because SoftWindows is designed to be a cross-platform solution, the technology has been ported to several different RISC architectures and platforms. After porting the core technology to a new architecture, FWB tunes it for optimum performance. The processor emulation design has evolved so that the performance features of today's latest RISC processors are harnessed. Moreover, each version of SoftWindows uses the best hardware and operating system features of a specific platform.

Networking

A particularly attractive feature of SoftWindows is the ability to provide nearly invisible PC networking functionality. It's based on the Network Driver Interface Specification (NDIS) and Open Datalink Interface (ODI) standards, enabling you to use multiple protocols simultaneously. It's very stable and gives the impression that you're running on real PC hardware, thus shielding the user from the intricacy involved in attempting PC networking in an emulated environment.

Although not every application is included, several solutions are built in to address a variety of networking software that may be running on a PC, including Novell's NetWare, Microsoft's Windows NT Server, Banyan's VINES, and UNIX file servers. Again, you can use more than one networking protocol at the same time.

Hardware

SoftWindows for Solaris attempts to fully emulate PC hardware to run PC operating systems and application software. This approach accommodates PC software that's written using standard APIs, or software that bypasses the operating system and directly addresses the PC hardware.

SoftWindows has built-in support for multiple floppy drives using a host 3.5-inch drive, and uses the Solaris files as the PC C: drive. It can use Solaris directories, including NFS mounts, through SoftWindows SFA (File Sharing Architecture). Further, it uses the host serial port as the standard communication port, which can be assigned to any of four communications ports (COM1 through COM4), each having a selectable baud rate, parity, and word format. The software also includes support for three standard printer ports: LPT1 through LPT3.

A variety of input devices are also supported, including emulation for 101-key keyboards, and the mouse moves easily between PC and Solaris windows. Finally, PC format CD-ROMs can also be read from within SoftWindows.

Tested applications

We tested a variety of applications on SoftWindows with very little difficulty. It easily handled industry standard office applications from vendors like Microsoft and Corel. In an attempt to see exactly how much it would handle, we also tried graphics applications such as CorelDraw and JASC Paint Shop Pro. Again, they executed without problem, but their performance seemed a little lacking. Overall, it did a fine job of running all of the applications we threw at it.

This isn't to say that it can run *everything*, but it does appear adequate to run most applications. The newest version of SoftWindows includes complete emulation of Windows multimedia features, including support for DirectX. We were able to run multimedia formats like AVI files, although we didn't thoroughly test support for DirectX.

SoftWindows for Solaris includes Microsoft Internet Explorer, allowing you to browse the Web with support for a variety of plug-ins, including Java, ActiveX, RealAudio, Macromedia Shockwave, Adobe Acrobat and a wide variety of additional plug-ins.

We also tested the ability to share applications over a network, and again were very pleased with the results. We successfully shared files and exchanged information with real PC hardware without a single problem. Unlike previous versions, long filenames are now preserved between the two platforms.

Conclusion

SoftWindows for Solaris now emulates the Intel Pentium processor, which enables your Solaris workstation to run a large variety of Windows and MS-DOS applications. If you're looking for an easy solution to implement Windows emulation on your Solaris system, you should definitely look at SoftWindows from FWB. ✳

DOWNLOAD ftp.elementkjournals.com/sun/nov00

Newsbot cleans up

by Don Kuenz

The InterNetNews (INN) package comes with a handy tool named Newsbot that helps you maintain order in the wild world of usenet. You can use Newsbot to remove abusive articles from your newserver. For example, most people would classify articles generated by hipcrime (a spamming agent) as abusive. In that spirit, we'll show you how to use Newsbot to remove hipcrime articles from your local news server.

In order to use Newsbot, you need to run INN on your host. You also need GNU's gnumake and gcc compiler. You can pick up both GNU tools from http://sunfreeware.com.

In this article, we'll show you how to use Newsbot with a news server installed in the default /usr/local/news INN directory. Your news server also needs to use a traditional INN spool structure. In other words, INN stores article 12345 of the group named comp.unix.solaris in a file named /usr/local/news/spool/articles/comp/unix/solaris/12345. If your INN installation uses something other than the default directory, or if it uses nontraditional storage, you'll need to slightly change these directions.

Installing Newsbot

You can find Newsbot's source in a file named contrib/newsbot under the root directory of the INN's source distribution. Create a new directory to hold your Newsbot source, and then copy contrib/newsbot to the new directory. The Newsbot development team distributes their source as a shell archive or shar.

Use your favorite editor to open contrib/newsbot and you'll see that it's a plain text file that contains several source files embedded within sh and sed commands. Delete all of the lines above the first comment line, which begins with a pound sign (#). Save the remaining lines to a file named *newsbot.shar*, and then exit your editor. Extract files from newsbot.shar with the following command:

Listing A: *The required changes to Newsbot's Makefile*

```
# Your news spool
PATHSPOOL   = /usr/local/news/spool
# Where you want to put newsbot config file
PATHCONF = /usr/local/news/etc/newsbot.conf
# Where you want to put newsbot pattern files
PATHPATS = /usr/local/news/etc/newsbot.pattern
# Facility for syslog.
LOGFAC      = LOG_LOCAL6

        .
        .
        .

# FreeBSD, NetBSD, Linux
#
LDFLAGS  =
INCL  =
CC = gcc
CFLAGS   = -Wall -g
```

```
sh newsbot.shar
```

This command extracts the following files: a source file named newsbot.c, a GNU Makefile, some text files that contain documentation and several sample configuration files. Newsbot names its default configuration file newsbot.conf.

You need to change some lines in your Makefile before you build Newsbot. **Listing A**, on the previous page, shows what you need to change in your Makefile. In addition to changing path names, you also need to uncomment the lines shown at the bottom of **Listing A**, which pertain to Linux. Fortunately, the Linux build also works on Solaris.

After you finish making the changes shown in **Listing A**, you can build the Newsbot binary using the following command:

```
gnumake newsbot
```

Although the compiler displays warnings about invalid casts during the build, it should create a binary named newsbot. Use the following commands to install the newsbot binary:

```
cp newsbot /usr/local/news/bin
chown news /usr/local/news/bin/newsbot
chgrp news /usr/local/news/bin/newsbot
md /usr/local/news/etc/newsbot.pattern
```

Those commands copy your Newsbot binary into INN's default binary directory. They also create a directory to hold your pattern files. We won't actually use the pattern directory in our example, but we'll create it anyhow in case you decide to use it later. After you finish installing Newsbot, you need to configure it.

Configuring Newsbot

Newsbot and INN both contain a lot of powerful functionality. As you might suspect, configuring both demands a meticulous attention to detail. One small mistake and your server could start spewing out thousands of bad articles. You need to use extreme caution and carefully verify and debug all changes that you make to configuration files.

Listing B: *The contents of the INN configuration file named newsfeeds*

```
## Default of  everything to everybody.
ME\
        :*\::

# An off-line feed sent with innxmit.
news.isp.net/news.isp.net,localcancel\
        :*,!control.cancel,!control,!junk,!local.*\
        :Tf,Wnm:news.isp.net

# Newsbot feed
newsbot.funnel!:*:Tm:NEWSBOT!
NEWSBOT!:!*:Tc,W*H:\
  /usr/local/news/bin/newsbot -a -f \
  -C /usr/local/news/etc/newsbot.conf \
  -D /usr/local/news/etc/newsbot.pattern
```

You can do a couple of things to limit your exposure. First, use the following command to create a local.test group on your news server:

```
Ctlinnd newgroup local.test
```

If you keep articles posted to local.test out of your feeds, you can then experiment, and if things go badly, in most cases you'll limit the damage to your local news server.

The second safeguard involves spooling articles to a file first, and later feeding them to your downstream sites by explicitly invoking `innxmit`. That way, you can intervene as necessary. We use both safeguards in our configuration. Newsbot stores configuration information in the following two files:

```
/usr/local/news/etc/newsfeeds
/usr/local/news/etc/newsbot.conf
```

Listing B shows the contents of our newsfeeds file. You need to change the news.ips.net host shown to one of your own downstream hosts. As we said earlier, we tell INN to create files for that feed by specifying the `:Tf` option. Newsbot directly uses the next two feeds, `newsbot.funnel!` and `NEWSBOT!`. As you might suspect, `newsbot.funnel!` funnels all incoming articles into `NEWSBOT!`, which works with the newsbot binary to remove abusive articles.

Newsbot uses newsbot.conf to tell it how to identify abusive articles, and what actions to take after it spots one. In our example, we use Newsbot to locally cancel all articles that originate from a software application known as hipcrime. **Listing C** shows the contents of our newsbot.conf file.

Our configuration tells Newsbot to search for the word *hipcrime* in either the `From:` header or the `NNTP-Posting-Host:` header. When it finds such an article, it will locally cancel it by using the `ctlinnd cancel <Message-ID>` command. Let's see how all of the pieces fit together.

First, you need to define the `NNTP-Posting-Host` header by inserting an `HNNTP-Posting-Host` line. Newsbot already knows about the `From` header. Next, you enter the following two lines to tell Newsbot to search for the word hipcrime in both headers:

```
~ie killFrom   From               hipcrime
~ie killHost   NNTP-Posting-Host  hipcrime
```

Listing C: *Our Newsbot configuration, which we store in a file named /usr/local/news/etc/newsbot.conf*

```
# $Id: binkill.conf,v 1.3 1996/09/17 20:34:13 pb Exp $
#
# To configure this to your system, copy this file as newsbot.conf,
# then look for and adapt the following strings:
#
#       USER@I_NEED_TO_CONFIGURE_NEWSBOT
#               (for Approved:, X-Canceled-By: and possibly From:)
#               -> a mail address to reach you
#
#       USER    (end of Path: for generated cancels)
#               -> something nice. 'not-for-mail' if you like.
#
#       rnews    -> path to your rnews
#       uuencode -> path to your uuencode
#       sendmail -> path to your sendmail
#       Mail     -> path of some mail user agent
#
#       newsbot! -> feed name in your 'newsfeeds' file
#
# Customize header fields to keep track of
# (used in header matching)
# Builtin (already known) headers are:
#
#       From
#       Subject
#       Newsgroups
#       Message-Id
#       Date
#       References
#       Sender
#       Reply-To
#       Approved
#

###################
#
# We define a custom header field.
#

HNNTP-Posting-Host

###################
#
# Pattern matching on article headers
#
# This looks for the word hipcrime anywhere within the
# From header, regardless of case.
#
~ie killFrom From   hipcrime
~ie killHost NNTP-Posting-Host          hipcrime


###################
#
# Patterns (as used in actions)
# Patterns not declared here are read from files with the
# same name in the pattern directory.
#
# First line begins with 'P' followed by the pattern name
# Following lines begin with '='.
#
# The following pattern generates a ctlinnd cancel string, which
# we will feed into a shell to automatically cancel articles
# from hipcrime.
#

Phip.cancel-local
=/usr/local/news/bin/ctlinnd cancel '%header-message-id%'

###################
# Actions
#
# A ">" in front of "pipe to" means to append to the
# indicated file.
# actions                 pattern file            pipe to
#

# create a ctlinnd cancel script and send it to /bin/sh
Ahip.cancel-local   hip.cancel-local    /bin/sh

###################
#
# Checks over *
#

Fnewsbot.funnel!

###################
#
# Link conditions with actions.
#

I~killFrom      hip.cancel-local
I~killHost      hip.cancel-local
```

Both of these lines contain three columns separated by tabs, not spaces. You must use tabs; otherwise, Newsbot quietly ignores both lines, and it will take you a long, long time to figure out what's wrong.

The `~ie` in the first column tells Newsbot to search headers, ignore case and use regular expressions. The `killFrom` and `killHost` strings, which follow `~ie` in the first column, connect a search to an action. We'll describe actions later on. The second column, that contains `From` and `NNTP-Posting-Host`, tells Newsbot which headers to examine. The third column contains the search word, or regular expression, that we seek.

The next configuration entry in Listing C specifies a template with the following two lines:

```
Phip.cancel-local
=/usr/local/news/bin/ctlinnd cancel
➥'%header-message-id%'
```

Unfortunately, Newsbot literature refers to this template as a pattern, even though it has little, if anything, to do with the matching pattern we just mentioned. We use the word *template* in this article to keep the two concepts separate. We plan to locally cancel abusive articles by invoking `ctlinnd cancel '<Message-ID>'` because it only cancels articles on our local news server. That gives us a margin of error should something go wrong. At worst, we'll just drop articles from our own local news server. Newsbot fills in `%header-message-id%` with the actual `Message-ID` from the abusive article when it performs an action.

The next configuration entry in newsbot.conf defines an action with the following line:

```
Ahip.cancel-local   hip.cancel-local   /bin/sh
```

This entry contains three columns separated by tabs, not spaces. You must use tabs; otherwise, Newsbot quietly ignores the line. The first column, which contains `Ahip.cancel-local`, names the action. You can use any name you wish. The second column contains the name of a template, in this case `hip.cancel-local`. The third column, `/bin/sh`, specifies a binary that Newsbot should invoke with the action. When Newsbot does this action, it fills in the template, invokes the binary, and pipes the resulting template into the binary.

The next configuration entry specifies our INN feed with the following line:

```
Fnewsbot.funnel!
```

The name of the feed that you specify here must match the name of the feed that you specify in your `newsfeeds` file.

Listing D: *The required entries in /etc/syslog.conf that enable Newsbot to log its activities to /usr/local/news/log/newsbot.log*

```
#
# newsbot logging
#
local6.debug     /usr/local/news/log/newsbot.log
```

The final configuration entry connects patterns to actions by using the following lines:

```
I~killFrom      hip.cancel-local
I~killHost      hip.cancel-local
```

This entry contains two columns separated by tabs, not spaces. Again, you must use tabs; otherwise, Newsbot quietly ignores both lines. The first column contains the letter `I`, followed by a matching pattern that we specified at the top of the file. The second column contains an action. When Newsbot detects a match, it performs the action. After you finish configuring those two files, you can start using Newsbot.

Running Newsbot

Newsbot contains powerful functionality that effects articles posted to public forums. Many people become very emotional about what they post to usenet. Prepare to face public wrath if you allow Newsbot to cancel even a single article without justification. As the Newsbot documentation warns, "you will bitterly regret it."

With such high risks, you absolutely must diligently monitor Newsbot's activities, especially after you change any INN or Newsbot configuration files. You also want to refrain from generating control cancels until you thoroughly understand the impact of every single line in newsfeeds. We won't cover control cancels in this article because they are so prone to leakage.

By default, Newsbot logs its actions to the syslog facility named LOCAL6. You need to make sure that LOCAL6 correctly functions before you start Newsbot. Listing D shows you how to change /etc/syslog.conf to make syslogd route LOCAL6 messages to a file named newsbot.log. As you can see, Listing D shows two columns separated by tabs, not spaces. You must use tabs; otherwise, syslogd quietly ignores the lines. After you add the LOCAL6 line to syslog.conf, restart syslogd with the following two commands:

```
/etc/rc1.d/K40syslog stop
/etc/rc1.d/K40syslog start
```

Whenever you alter your newsfeeds file, you tell innd to start using the new configuration by invoking `ctlinnd reload newsfeeds xxx`. Whenever you alter your newsbot.conf file, you must restart the newsbot binary by invoking `ctlinnd begin NEWSBOT!`. Then, when the newsbot binary restarts, it enters a log entry into newsbot.log. You need to make sure you see that entry.

At this point, the `ps -e | grep newsbot` command should display a newsbot process. Post an article to local.test that contains the word *hipcrime* in the `From` header to make sure that Newsbot correctly cancels it. Keep an eye on the articles appended to the file named /usr/local/spool/outgoing/news.ips.net, which you'll later feed to your downstream feed by using the `innxmit` command. Make sure that all of the log entries in newsbot.conf look proper.

Conclusion

It's a fact of life that people post abusive articles to usenet. Newsbot can help you automatically process such articles at your discretion. To summarize, you simply need to extract the Newsbot source distribution from a shar archive, and then configure newsfeeds and newsbot.conf. Following the steps in this article will help you use Newsbot. ✳

DOWNLOAD ftp.elementkjournals.com/sun/nov00

Configuring BIND 8: Part 2

by Jerry L.M. Phillips

In the August 2000 issue of *Inside Solaris,* "Configuring BIND 8" discussed domain names, BIND's architecture, and how to configure BIND 8 to use it as a DNS client and DNS server. In this article, we'll show you how to configure your BIND 8 Domain Name System (DNS) server to accommodate additional domains.

Adding more domain names

In the previous article, we first configured a DNS server for the inside.biz domain. What if your business, which previously registered the inside.biz domain, decides to also register one or more additional domains and use the same DNS server as well? Again, you can use the search engine at **http://rs.internic.net/whois.html** to determine the uniqueness of the additional domain names. Then, you can pick a registrar from **http://rs.internic.net/alpha.html**.

In this example, you register inside.com, inside.org and inside.net. As part of the registration process, and following the August article's example, you specify that the primary DNS is *apollo.inside.biz* and its address is *192.42.172.130*.

Reconfiguring the DNS server

Listing A shows the /etc/named.conf file with the three new additional domains (or zones). Note that each new domain refers to the db.inside file in order to find hostname-to-IP address maps for your inside.com, inside.org and inside.net domains.

Listing B, on the next page, shows the contents of the reconfigured db.inside file. Your business is growing fast, so each of the new domains will have a separate Web server. Note the relevant comments and accompanying statements in the db.inside file that define the individual Web

Listing A: *Contents of the /etc/named.conf file*

```
Options {
   ///etc/named.conf
   //
   //boot file for primary name server
   //
   //type domain source file or host
   //
   directory  "/var/named";
};
zone "inside.biz" in {
   type master;
   file "db.inside";
};
zone "inside.com" in {
   type master;
   file "db.inside";
};
zone "inside.org" in {
   type master;
   file "db.inside";
};
zone "inside.net" in {
   type master;
   file "db.inside";
};
zone "172.42.192.in-addr.arpa" in {
   type master;
   file "db.192.42.172";
};
zone "0.0.127.in-addr.arpa" in {
   type master;
   file "db.127.0.0";
zone "." in {
   type hint;
   file "named.ca";
};
```

Listing B: *Contents of the /var/named/db.inside file*

```
; db.inside - hostname to IP address resolution table
@  IN SOA  apollo.inside.biz.    root.apollo.inside.biz. (
      97011001  ; Serial number
      10800     ; Refresh after three hours
      3600      ; Retry after one hour
      604800    ; Expire after one week
      86400 )   ; Minimum TTL of one day

   IN NS    apollo.inside.biz.

; Define the localhost
localhost  IN A 127.0.0.1
; Define the hosts in this zone
apollo.inside.biz.        IN A 192.42.172.130
aphrodite.inside.biz.     IN A 192.42.172.132
zeus.inside.biz.          IN A 192.42.172.136
hyperion.inside.biz.      IN A 192.42.172.141
eos.inside.biz.           IN A 192.42.172.145
hermes.inside.biz.        IN A 192.42.172.146

; Define web server in inside.com domain
www.inside.com.  IN A 192.42.172.147
; Define web  server in inside.org domain
www.inside.org.  IN A 192.42.172.148
; Define web server in inside.net domain
www.inside.net.  IN A 192.42.172.149

; Add CNAME records, as desired (for host aliases)
;loghost IN CNAME apollo.inside.biz.

; Add MX records (mail exchangers) below
inside.biz.IN MX 0 apollo.inside.biz
inside.com.IN MX 0 www.inside.com.
inside.org.IN MX 0 www.inside.org.
inside.net.IN MX 0 www.inside.net.
```

Listing C: *Contents of the /var/named/db.192.42.172 file*

```
@  IN SOA   apollo.inside.biz   root.apollo.inside.biz. (
      97011001 ; Serial number
      10800    ; Refresh after three hours
      3600     ; Retry after one hour
      604800   ; Expire after one week
      86400 )  ; Minimum TTL of one day

   IN NS   apollo.inside.biz.

130.172.42.192.in-addr.arpa.  IN PTR apollo.inside.biz.
132.172.42.192.in-addr.arpa.  IN PTR aphrodite.inside.biz.
136.172.42.192.in-addr.arpa.  IN PTR hyperion.inside.biz.
141.172.42.192.in-addr.arpa.  IN PTR zeus.inside.biz.
145.172.42.192.in-addr.arpa.  IN PTR eos.inside.biz.
146.172.42.192.in-addr.arpa.  IN PTR hermes.inside.biz.
147.172.42.192.in-addr.arpa.  IN PTR www.inside.com.
148.172.42.192.in-addr.arpa.  IN PTR www.inside.org.
149.172.42.192.in-addr.arpa.  IN PTR www.inside.net.
```

servers. Also, note the MX records added for each of the individual mail servers.

It's possible that you can use one computer to function as a host for all three Web servers, but that's another subject altogether. In "Configuring BIND 8," we showed you that the DNS server uses another file, that is, db.192.42.172, to return a matching hostname whenever a client queries it with an IP address. We also discussed that this technique is known as reverse-mapping. **Listing C** shows the contents of the reconfigured db.192.42.172 file.

Conclusion

In this article, we've shown you how to register three new domains on the Internet and configure your existing DNS server to resolve IP address-to-hostname and hostname-to-IP address queries for individual Web servers on each of those domains. You can become very creative with domains/zones on a DNS server. So, experiment as much as you can. ✳

Cross-platform ASP

by Clayton E. Crooks II

Introduced in 1996, Microsoft's Active Server Pages (ASP) have altered the way developers look at client/server computing and application development on the Internet. Although a variety of alternative solutions exist, it appears Microsoft has been successful with their attempts at controlling this market as most developers view ASP as the predominant means for scripting server-side applications. Unfortunately, as with many Microsoft products, they limited ASP by developing it exclusively for the Windows platform and, in this instance, Windows Internet Information Server (IIS). However, with new offerings from companies like Chili!Soft and Halcyon, your ASP platform choices are rapidly escalating.

ASP clones

Chili!Soft's ASP and Halcyon Software's Instant ASP (iASP) are two ASP clones that were developed to run under non-Microsoft operating systems and Web servers. Halcyon and Chili!Soft deliver products that offer capabilities very similar to those offered by the true Microsoft implementation. Although compatibility issues can't easily separate them, they are very different solutions and many developers will probably prefer one to the other.

Both solutions offer a high degree of compatibility with ASP, although they take drastically different approaches in getting the job done. With support for a variety of platforms, companies can develop apps for ASP using Microsoft and a plethora of third-party tools while running them on other operating systems and Web servers.

Chili!Soft ASP

The first of the two ASP clones we'll look at is Chili!Soft ASP, which is definitely the more established of the two products. It was the first company to effectively port ASP and now, in version 3.x, it's a fairly mature product, albeit one that continues to have growing pains.

The software was developed using C++, which makes it difficult to port, so it has been rather slow to add additional platforms. It also appears that they have been very specific with the combinations they have certified to work with their software. If you're planning to use Chili!Soft, make sure to check for a particular combination.

One of the more interesting features of Chili!Soft ASP is the licensing of third-party software, such as the use of the Microsoft-licensed VBScript and JScript scripting engines. Having Microsoft's blessing is important, as they are attempting to clone the Microsoft-centered ASP.

Their relationships don't stop with Microsoft. Chili!Soft ASP also includes SQLink from Merant (**www.merant.com**), which provides a wide range of ODBC connectivity for various databases. You'll find that iASP also supports Merant, but you must purchase the drivers separately. Because of an additional vendor product (Mainsoft's Mainwin) and the COM support it provides, Chili!Soft ASP is compatible with ASP scripts that use ADO COM objects, FileSystem and BrowserCaps. In order to support ADO, Chili!Soft ASP includes UNIX ODBC drivers for a variety of accepted relational databases.

Although it does an adequate job of covering native Microsoft ASP, Chili!Soft ASP isn't a complete ASP implementation. It appears that Chili!Soft supports nearly all of the built-in components of ASP, but it lacks total ADO support. Moreover, it's sometimes playing catch-up with current version support. For instance, it currently supports version 3 of VBScript and JScript, although they do have licenses for the new version 5, which will be released shortly. Another area that's lacking is support for Microsoft Transaction Server (MTS).

Halcyon iASP

Halcyon is a more recent participant in porting ASP to other platforms. Its first product, Instant ASP (iASP), lets you use ASP in a variety of Web environments. You'll find that iASP, which was first released in the middle of 1999, runs on many platform combinations that Chili!Soft doesn't support. However, because it was released more recently, iASP still doesn't support as much of the functionality of ASP as Chili!Soft's product, although this gap is rapidly disappearing.

As we mentioned earlier, Chili!Soft ASP was written using C++. On the other hand, iASP was written in Java, which makes it portable to almost any operating system that has a Java runtime environment and a Web server with a Java servlet API. This, in turn, makes it much easier to port to additional platforms.

If your preferred development language is Java, iASP is obviously something to consider, as it offers a full suite of Java developer APIs. An added benefit is the ability to run alongside third-party application servers that have Java servlet support. It also supports Sun's competing JavaServer Pages technology.

Another difference between the products is the use of scripting languages. Halcyon wrote their own JScript and VBScript interpreters that are compatible with Microsoft's JScript and VBScript version 3.2. The development keeps them free of licensee restrictions and influence from outside sources. Halcyon has also written JavaBeans versions of most of the standard Microsoft ASP components like the Ad Rotator component, browser capabilities and the File Access component.

The fact that Halcyon developed their software with an approach that differs from those of Chili!Soft doesn't keep iASP from suffering from the same types of problems. ASP consists of a number of different objects—the `Server`, `Request`, `Response`, `Application` and `Session` objects—but they aren't implemented consistently by iASP across all platforms. Again, it would be a good idea to verify that the solution you're planning will completely work with your platform of choice.

Database connectivity is an important topic for ASP in general, and specifically when looking at clone options. iASP supports an ADO access object that conforms to the ADO 2.1 release from Microsoft. This lets you access both OLE DB and JDBC databases. As a Java application, iASP works only with Java Database Connectivity interfaces for databases. If you want to connect with Access or SQL Server databases, Halcyon recommends using the Merant database drivers, which are the same drivers that Chili!Soft bundles in its offering.

Conclusion

Chili!Soft and iASP aren't the only players in the ASP clone market, although they are the most advanced. An open source offering, Open ASP, available at **www.activescripting.com**, looks promising, but it appears that development may have stopped.

It's worth noting that both iASP and Chili!Soft ASP are complex pieces of software that depend on a variety of software components. Because of the extremely complicated relationships between these pieces of software, certification of the respective products is often limited to only a select few combinations of Web servers and operating systems. As a result, make sure to check the certification for the platform you intend to use.

Chili!Soft ASP and iASP each take drastically different approaches in their attempts to make ASP available on multiple platforms. With only a few exceptions, both work very similarly to the Microsoft implementation that they are based on. At this time, Chili!Soft is a more mature package and offers more complete compatibility with Microsoft's ASP. However, it costs more and runs on fewer platform variations, and sometimes is even limited by minor revision changes. Halcyon continues to add functionality to iASP, but it has not yet caught up with its predecessor.

Both companies offer downloadable versions (**www.chilisoft.com** and **www.halcyonsoft.com**, respectively) that you can use to evaluate them yourself. Not only is it a good idea to test them on a particular platform, but each of them offers enough compelling reasons to test them for yourself. ✳

SOLARIS Q & A

Renaming your Sun computer

by Jerry L.M. Phillips

I want to change the hostname of my Sun Microsystems computer. How do I accomplish that?

You can change the hostname the hard way or you can do it the easy way. The hard way is to execute the program /usr/sbin/sys-unconfig, which restores the system configuration to its original form at shipping time. (Read the sys-config man pages for more detail.) Then, sys-unconfig reboots the machine and steps you through the system configuration process using the sysidtool suite of programs. (Read the sysidtool man pages for more detail.) This is overkill when all you want to do is change the computer name! The easy way is to edit several files carefully and replace the existing name with your new name. Be cautious in what you choose for your host name. The files to edit are:

```
/etc/hostname.hme0 (or /etc/hostname.le0 if 10Mb)
/etc/nodename
/etc/inet/hosts
/etc/net/ticlts/hosts
/etc/net/ticots/hosts
/etc/net/ticotsord/hosts
```

The last three host files are referred to as loopback transport providers. Now, to make the hostname active, you can issue the command using your new hostname as the argument to the hostname program:

```
# /usr/sbin/hostname new_host_name
```

You should probably reboot to ensure that your new hostname works. During the reboot, you could possibly receive a message similar to the following:

```
Jul 26 09:44:23 superman sendmail[217]: My
➥unqualified host name (superman) unknown;
➥sleeping for retry
Jul 26 09:45:23 superman sendmail[217]:
➥unable to qualify my own domain name
➥(superman) -- using short name
```

In this example, your new hostname is superman. But, the sendmail daemon on your machine can't resolve the new hostname when talking with the DNS server as specified in your /etc/resolv.conf file. So, you'll have to alter your hostname on the DNS server in conjunction with this change. Also, if you experience a problem with your CDE mail utility, subsequent to your hostname change, you might look at your .mailcap file. If you find a `set smtpserver=` statement that contains your old hostname, you'll need to change it.

Virtual interfaces on Solaris

I am in the process of configuring a Web server to run multiple domains. How do I configure Solaris to accommodate several virtual interfaces?

This may take some explaining. First, you `use ifconfig -a` to see all of the current interfaces on your Web server platform. You'll also need to log on as a superuser in order to perform the following tasks. This yields two interfaces:

```
# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232 inet 127.0.0.1
➥netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS, RUNNING,MULTICAST> mtu 1500
➥inet 157.21.1.17 netmask ffffff00 broadcast 157.21.1.255 ether
➥8:0:20:ac:5e:a7
```

The two interfaces are lo0:, the local loopback, and hme0:, the interface to your ethernet network card. We will ignore lo0: in our example. The IP address for your Web server platform is 157.21.1.17. Let's say that you dedicate that interface to www.inside.com. Now, let's add another interface:

```
# ifconfig hme0:1 plumb
# ifconfig hme0:1 157.21.1.18
# ifconfig hme0:1 netmask 255.255.255.0
# ifconfig hme0:1 broadcast 157.21.1.255
# ifconfig hme0:1 up
```

Let's see how our interface configuration looks:

```
# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232 inet 127.0.0.1
➥netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING, MULTICAST> mtu 1500 inet
➥157.21.1.17 netmask ffffff00 broadcast 157.21.1.255
        ether 8:0:20:ac:5e:a7
```

About our contributors

Clayton E. Crooks II is a self-employed computer consultant living in Knoxville, Tenn. He's married with one child. His hobbies include game development, 3-D modeling and any athletic activity he can find time for.

Werner Klauser is an independent UNIX consultant working near Zurich, Switzerland. While not paragliding or roarin' around on his Harley chopper, he can be reached by email at **klauser@klauser.ch** or on his Web page at **www.klauser.ch**.

Don Kuenz works at Computing Resources Company (**http://gtcs .com/crc**). They provide programming, administration and hardware for Sun and PC platforms. You can reach Don at **kuenz@gtcs.com**.

Jerry L.M. Phillips, M.S. is director of the database center at Eastern Virginia Medical School. In addition to his administrative duties, he manages Sun/Solaris-based platforms for the medical school, including DNS, sendmail, WWW, anonymous FTP, proxy and library servers.

Inside Solaris™
Tips & Techniques for users of Sun Solaris

Inside Solaris (ISSN 1081-3314) is published monthly by Element K Journals, a division of Element K Press, 500 Canal View Boulevard, Rochester, N.Y., 14624.

Customer Relations

U.S. toll free	(800) 223-8720
Outside of the U.S.	(716) 240-7301
Customer Relations fax	(716) 214-2386

For subscriptions, fulfillment questions, and requests for group subscriptions, address your letters to

Element K Journals Customer Relations
500 Canal View Boulevard
Rochester, NY 14623

Or contact Customer Relations via Internet email at journals@element-k.com.

Editorial

Editor	Garrett Suhm
Assistant Editor	Jill Suhm
Managing Editor	Michelle Rogers
Assistant Managing Editor	Dianne Galloway
Copy Editors	Rachel Krayer Glenna Lechner
Contributing Editors	Clayton E. Crooks II Werner Klauser Don Kuenz Jerry L.M. Phillips
Graphic Designer	Rachel J. King
Cover and Content Design	Melissa Ribaudo

You may address tips, special requests, and other correspondence to

The Editor, *Inside Solaris*
500 Canal View Boulevard
Rochester, NY 14623

Editorial Department fax	(716) 272-0064

Or contact us via Internet email at inside_solaris@elementkjournals.com.

Sorry, but due to the volume of mail we receive, we can't always promise a reply, although we do read every letter.

Element K Journals

General Manager	Kelly Baptiste
Manager of Customer Relations	Nicole Pate
Manager of Operations	Cristal Haygood
Manager of Graphic Design	Ian Caspersson
Manager of Product Marketing	Mike Mayfield
Senior Product Marketing Manager	Brian Cardona

Postmaster

Periodicals postage paid in Rochester, N.Y., and additional mailing offices.

Postmaster: Send address changes to

Inside Solaris
P.O. Box 92880
Rochester, NY 14692

Copyright

Printed in the U.S.A.

Price

Domestic	$129/yr ($11.00 each)
Outside U.S.	$149/yr ($13.00 each)

Our Canadian GST# is: R140496720. CPM# is: 1446703.
QST# is: 1018491237.

Back Issues

To order a back issue from the last six months, call Customer Relations at (800) 223-8720. Back issues cost $11.00 each, $13.00 outside the U.S. You can pay with MasterCard, VISA, Discover, or American Express.

Are you moving?

If you've moved recently or you're planning to move, you can guarantee uninterrupted service on your subscription by calling us at (800) 223-8720 and giving us your new address. Or you can fax us your label with the appropriate changes at (716) 214-2386. Our Customer Relations department is also available via email at journals@element-k.com.

Coming up...

- Using NFS
- Tracking sessions with Truss

```
hme0:1: flags=843<UP,BROADCAST,RUNNING,MULTICAST>
➥mtu 1500 inet 157.21.1.18 netmask
➥ffffff00 broadcast 157.21.1.255
```

The virtual interface hme0:1 has appeared. Let's say that you dedicate that interface to www.inside.org. Now, let's add another interface:

```
# ifconfig hme0:2 plumb
# ifconfig hme0:2 157.21.1.19
# ifconfig hme0:2 netmask 255.255.255.0
# ifconfig hme0:2 broadcast 157.21.1.255
# ifconfig hme0:2 up
```

Let's see how our interface configuration looks again:

```
# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST>
➥mtu 8232 inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,
➥RUNNING,MULTICAST> mtu 1500 inet
➥157.21.1.17 netmask ffffff00 broadcast
➥157.21.1.255 ether 8:0:20:ac:5e:a7
hme0:1: flags=843<UP,BROADCAST,RUNNING,
➥MULTICAST> mtu 1500 inet 157.21.1.18
➥netmask ffffff00 broadcast 157.21.1.255
hme0:2: flags=843<UP,BROADCAST,RUNNING,
➥MULTICAST> mtu 1500 inet 157.21.1.19
➥netmask ffffff00 broadcast 157.21.1.255
```

The virtual interface hme0:2 has appeared. Let's say that you dedicate that interface to www.inside.net.

So, what have you done? The initial virtual interface was hme0:0. (It defaults to hme0: on the output from the `ifconfig -a` command. The zero after the colon is the logical unit number.) That interface will respond to queries for 157.21.1.17.

The first interface that you added manually was hme0:1, which responds to queries for 157.21.1.18. The second interface that you added manually was hme0:2. That interface responds to queries for 157.21.1.19.

Prior to these additions, you should have configured your DNS server to handle the hostname-to-IP address and IP address-to-hostname translations for you. Note that Solaris supports 255 virtual interfaces in the form of logical units, i.e., 0,1,2,...254. Make sure you can ping both the hostnames and IP addresses, which you set up, from other computers on your network. Finally, add the `ifconfig` commands to a run control script that will activate the virtual interfaces every time you boot the Web server platform. You can put the statements in a file such as /etc/rc2.d/S98ifconfig. ✳

QUICK TIP

Looking out for setuid programs

A *setuid* program lets a user run the program with the permissions of the program owner. For example, if you create a setuid program, with the owner root, any person that runs the program will do so with root user permissions. You need setuid programs for many applications, but they also create security problems.

In a recent break-in of an Internet service provider, hackers broke into a system and stayed logged on to it as the root user just long enough to create a setuid program. When they logged into the system later as another user, they used their setuid program to achieve their mischief.

As an administrator, you want to keep a look out for setuid programs. An easy way to do this is by using the `find` command with the `-perm` option. Here's how to find every setuid program on your system, and generate an `ls` listing of those programs:

```
find / -perm -4000 -exec ls -ld {} \; >
/tmp/suid.files
```

This searches your filesystem for all setuid programs, and prints the listing to the file named /tmp/suid.files. There are a fair number of setuid files on a basic Solaris system, so the key is to look at how this listing changes from day to day, week to week, or month to month (depending on your security concerns). ✳