



Inside Solaris™

Tips & Techniques for users of Sun Solaris

The Sun still rises

As we look toward the first actual year of the millennium, it's fun to look behind us to see where we've been and try to guess where we're going. Without a doubt, it's been a wild ride for Sun and its flagship operating system Solaris. Let's start by grading Sun, Solaris and the industry in general.

The OS challenge— Sun makes the grade

The year 2000 was supposed to be the year of the challengers of Sun's almost complete dominance of the mid-range and high-end server solutions market. From every direction, Sun was supposed to lose significant ground from up-and-coming competitors.

From one corner, Microsoft's release of Windows 2000 should have (at least according to the pundits) taken over the world at the high end like Windows did on the desktop. But the world of business had no intention of trusting their mission-critical servers to a proprietary operating system that hadn't been battle-tested in 24x7 environments.

From the open-source corner, Linux was also supposed to destroy everything in its path. But the combination of scaling problems (even with four processors) and the immaturity of many commercial products in this space (Veritas and Oracle, for example) also kept Linux from making a big dent in Solaris. If anything, most Solaris administrators are using Linux for departmental solutions, developer's desktops and mail servers.

Sun's grade: A

Hardware

Sun has had an incredible run this year, not only in the stock market, but with customers as well. But the current generation UltraSparc II was starting to show its age, leaving Sun open to challengers with faster systems. This year has been a major dose of *deja vu* for us. Flash back to 1996—Sun was dominating with their excellent Sparc 1000 and 2000 Enterprise servers, as well as Sparc 10 and 20 workstations. But they were quickly getting passed by in the horsepower race by DEC (remember them?) and the speedy Alpha chips running at around 400 MHz.

Sun's only defense was the fact that they had a lot more processors per box (eight for the Sparc 1000, and even more for the 2000) and were relatively less expensive per processor. But they were running at a pokey 70 or 80 MHz, and Sun was starting to look like they were falling behind in the market. Then the revolutionary first-generation UltraSparc was introduced along with the line of Enterprise-class servers like the 3000, 4000 and 6000. Sun proved you could build modular systems that could be upgraded by simply getting a larger chassis and using your existing components.

As living proof of the power of this architecture, we built a system for a Fortune-500 company that started life as a two-way E4000, and is now an E6000 with more than 12 processors and over two terabytes of storage. This type of power and the explosion of the Internet have helped propel Sun to where they are today.

In This Issue

The Sun still rises

Tuning Solaris
for FireWall-1

Security: threats
and mistakes

OCR for Solaris

Exploring REGEXs
with grep

Inside Solaris
2000 index

Now we're at the close of the year 2000 with the same situation. Luckily for Sun, the competition has been having its own problems. The Intel Merced/Itanium chip was supposed to rule the world, but is a day late and a dollar short. Most of the competition has come from IBM and their S series boxes.

So what has Sun done this time to avoid being totally bypassed in the CPU wars? They have finally released their much-delayed UltraSparc III in an incredible package—the Sun Blade 1000. This system can handle up to two 900 MHz UltraSparc III processors (with an incredible 8 MB of cache), 64-bit PCI, and a 4 GB/second system interconnect. They also added USB and Firewire interfaces for high-speed peripheral connection, and internal Fibre-channel drives.

While this is quite impressive, they'll need to spread the UltraSparc III quickly throughout their whole Enterprise server line if they want to remain competitive. Especially at the high end, their aging E1000 servers are in danger of being trumped by nimble, high-performance IBM clusters, and maybe even HP's recently announced Superdome.

At the workstation level, Sun's Ultra 5s and Ultra 10s have proven that workstations can be built at good prices. This has probably done more than anything to help cultivate the strong Sun market, allowing developers and administrators the chance to have their own Sparc-based Solaris "sandbox" in which to play. Previously, this could only be done with Solaris x86 and its much smaller base of software.

On the low end, Sun has re-entered the network computer market after a failed first attempt. The Sun Ray provides a compelling non-PC platform that has tremendous potential in places where standard PCs have been proven to be costly and hard to manage. The integrated Smart Card technology is a no-brainer in places where machines are shared by multiple people (think anywhere with shifts, such as a nurses' workstation). Flat-panel screens and larger displays, as well as different form factors, make this a much more flexible solution than Sun's first try. However, the need for a dedicated 100 Mb connection to a Sun server can prove to be costly.

Sun's grade: B

Open source

It's been a rocky road for Sun on this one. Most of the problems have centered around Java, which

happens to be one of the biggest uses of Sun and Solaris in Enterprise. While they don't have the same reputation for proprietary closed-mindedness as Microsoft, they have gotten beaten up heavily for pulling out of the International Standards Organization and the European Manufacturer Computer Association (ECMA) open standards process for Java. Many believe that this proves that Sun had no real desire for open standards, but we feel that the ECMA wasn't the best forum for Java standards anyway.

So where did that leave us this year? Sun tweaked the Java community process with version 2. This forum provided an improved mechanism for incorporating changes to the Java platform. But it still falls short of what most people would like to see out of the Java platform.

Sun's grade: C

Software

The Internet has been good to the growth of the Solaris platform. Almost all UNIX applications written today are available on the Solaris platform. With the incredible growth of the Solaris and Linux platforms, the sheer volume of applications and solutions are amazing.

Even though Microsoft has a lock on the desktop Office suite segment, Sun's StarOffice has been downloaded over 3 million times at the time of this writing.

Java has helped fuel Sun's growth and will continue to in the future. With a temporary setback on the desktop, Java has found a home on the server with every major vendor (except Microsoft) offering server-side solutions. Sun's Java 2 Enterprise Edition (J2EE) technologies are becoming the de facto standard for Internet-delivered applications. Enterprise Java Beans (EJBs) have been doing for middleware what Corba has been promising for years. Even the iPlanet Java server has made a comeback after being swallowed by the Netscape/AOL/Sun alliance.

Sun's grade: A

Solaris 8

This year saw the release of the evolutionary Solaris 8, once again proving that Sun understands the meaning of *mission critical*. Solaris 8 is a robust and scalable platform that doesn't operate like a certain large proprietary monolith also released this year. And to make things even sweeter, you can now download Solaris from Sun or get a CD for a nominal charge. And you can use Solaris at home or work on any machine with up

to 8 CPUs for free. You can even get the source code for Solaris 8.

New features also make Solaris 8 a compelling upgrade. Solaris 8 comes with IPv6 support, Kerberos authentication, and better cluster support.

Sun's grade: A

The Future

So it's been a great year for Sun and Solaris. Overall, we'd give them a solid B+ with their strategy. But what's going to happen next year? Will the market continue to be great for everyone who is affected by Solaris? Will Linux or Windows 2000 take over the world?

We think if Sun keeps rolling out UltraSparc III-based systems throughout their product line, the future will be bright. They have proven that their singular focus on UNIX and Solaris was the right move in the past, and we think it will continue to be the right move for the future. Compared to the garbled message of companies like HP (Windows one day, HP/UX the next), Sun has done a great job of staying the course.

So how about the *Inside Solaris* journal? What does the future hold? Expect us to continue to offer more of the same—answering your questions about security, configuration, administration, and tuning, as well as providing reviews of selected software that helps make your life easier. *

Tuning Solaris for FireWall-1

by Rob Thomas

Sun and CheckPoint have an intertwined history. Long resold as Solstice FireWall-1, it has become one of the most ubiquitous firewall packages in use today. While FireWall-1 on Solaris makes for a strong bastion, Solaris isn't inherently tuned to provide filtering and routing services. In this article, we'll address some of the steps necessary to create an efficient and robust firewall.

Introducing FireWall-1 on Solaris

CheckPoint FireWall-1 runs in kernel space on the Solaris platform. As shown in **Figure A**, by inserting itself between OSI layers 2 and 3, CheckPoint captures and inspects all of the packets before the IP input routines (e.g., `ipintr()`) are called. If the CheckPoint rule base denies a packet, it will be dropped before traveling further up the stack.

If the packet is allowed, it's passed further up the stack for processing. This processing is usually the routing of the packet out of another interface on the Solaris firewall.

While Solaris includes a robust IP stack, it isn't a platform built strictly for routing. Therefore, a fair bit of tuning is required to enhance the routing capabilities of Solaris. Regardless of the rule base, a FireWall-1 host is a router above all else, and you should be tuned accordingly. It's important that the packets spend as little time as possible inside the firewall.

Also note that the hardware platform makes a significant difference to the overall performance of the firewall. Our tests have shown that a Sun E450, dual 200 Mhz CPUs, 1 GB RAM, a single QFE card, running Solaris 2.6 and CheckPoint FireWall-1 4.0 with a rule base containing 258 rules will provide throughput of 94.50 Mb/s without load, and throughput of 39.49 Mb/s with a 100 Mb/s load presented on two of the four interfaces. If your throughput requirements are considerably above these numbers, then you should consider the Nokia platform, specifically the Nokia IP650. You can find more information about the Nokia firewall platform at www.nokia.com/securitysolutions/.

Tuning the IP stack with `ndd`

We'll conduct our first set of tunings using the `ndd` command. The `ndd` command, found in `/usr/sbin` under Solaris 7, allows you to peruse and modify the settings of the IP stack.

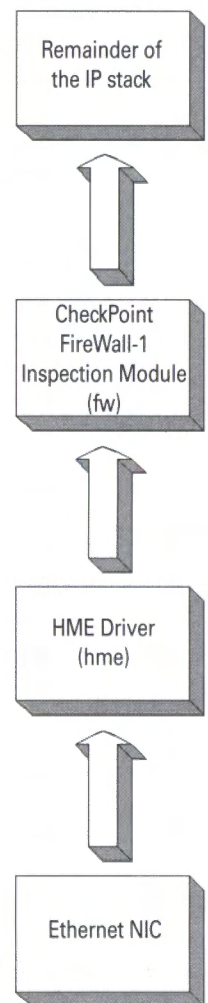


Figure A: This is the integration of CheckPoint into the IP stack.

These tunings will provide enhanced performance and a robust defense against the malcontents.

You should place these modifications in `/etc/init.d/inetinit`. We recommend placing them at the top of the file, surrounded by detailed comments. You can also run these commands from the command line. **Listing A**, complete with comments, shows the initial settings.

If you're logging the firewall to a remote host (e.g., a remote Enterprise Management Console), then you should include additional enhancements. Depending on the rule base, the logging volume can be quite copious and create a bevy of TCP traffic between the firewall and the EMC. It's important to increase the efficiency of this TCP traffic, and you can accomplish this by enabling

both RFC1323 and RFC2018 on both the firewall and the EMC. If the EMC host doesn't support RFC1323 and RFC2018, don't enable them on the Solaris firewall. Also include these modifications in `/etc/init.d/inetinit`:

```
# Increase the efficiency of TCP connections
# See RFC1323 and RFC2018 for more information
# Support SACK, RFC2018
/usr/sbin/ndd -set /dev/tcp tcp_sack_permitted 1
# Go with larger send and receive windows, RFC1323
/usr/sbin/ndd -set /dev/tcp tcp_xmit_hiwat 65535
/usr/sbin/ndd -set /dev/tcp tcp_rcv_hiwat 65535
```

Once you complete the IP stack tuning, you're ready to modify the STREAMS queues.

NOTE:

In this article, we assume that you're familiar with the operation of CheckPoint FireWall-1, as well as the basics of TCP/IP. For further information on FireWall-1, visit www.checkpoint.com or Phoneboy's FireWall-1 site at www.phoneboy.com/fw1/. For a deeper understanding of TCP/IP and the implementation of TCP/IP on UNIX, peruse *TCP/IP Illustrated, Volume 1* by W. Richard Stevens and *TCP/IP Illustrated, Volume 2* by Gary R. Wright and W. Richard Stevens.

Listing A: Initial settings for our Checkpoint Firewall configuration

```
# Do not forward directed broadcasts, e.g. Smurf attacks
/usr/sbin/ndd -set /dev/ip ip_forward_directed_broadcasts 0
# Do not forward source routed packets
/usr/sbin/ndd -set /dev/ip ip_forward_src_routed 0
# Do not respond to queries for our netmask
/usr/sbin/ndd -set /dev/ip ip_respond_to_address_mask_broadcast 0
# Do not respond to broadcast ICMP_ECHO (ping)
/usr/sbin/ndd -set /dev/ip ip_respond_to_echo_broadcast 0
# Do not respond to broadcasted timestamp queries
/usr/sbin/ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
# Do not issue redirects -- fix the routing table instead
/usr/sbin/ndd -set /dev/ip ip_send_redirects 0
# Don't let others modify our routing table
/usr/sbin/ndd -set /dev/ip ip_ignore_redirect 1

# Increase our defense against SYN floods
# The "q" queue is the completed socket holding pen where
# sockets remain until the application issues accept()
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q 1280
# The "q0" queue is the half-open socket queue
/usr/sbin/ndd -set /dev/tcp tcp_conn_req_max_q0 10240
```

Modifying the STREAMS queues

The Solaris 7 IP stack is based on STREAMS. This paradigm presents significant improvements and performance benefits. We'll make only one enhancement here. This enhancement requires a bit of calculation. To determine the optimal size of the STREAMS synchronized queues, multiply the total physical RAM divided by 64 25 times. For example, 64 MB of RAM will yield 25, as in $25 * (64/64)$. 128 MB of RAM will yield 50 as in $25 * (128/64)$.

Now, insert the following (with your result in place of the number 25) in `/etc/system`. Remember, modifications to `/etc/system` require a reboot to install:

```
* Increase STREAMS queues
set sq_max_size = 25
```

Modifying the behavior of fsflush

On firewalls, it's common to have quite a bit of physical memory. This is particularly helpful as we previously noted. However, as the amount of physical memory is increased, the amount of time the kernel spends managing that memory also increases. During periods of high load, this may decrease throughput.

To decrease the amount of memory fsflush scans during any scan interval, we must modify the kernel variable `autoup`. The default is 60. For firewalls with 128 MB of RAM or more, we shall increase this value by placing the following in `/etc/system`:

```
* Decrease the amount of RAM scanned by fsflush
set autoup = 120
```

The end result is less time spent managing buffers and more time spent servicing packets.

Blackholes

Even the most robust security screens suffer from some weakness, and it's possible that someone will breach your network. These attacks may take the form of spoofed sources hitting your hosts, resulting in your end stations attempting to respond to these spoofed addresses. Fortunately, the script kiddies often use bogon networks, or networks that aren't officially allowed on the Internet. We can utilize black hole routes to block traffic to these bogon networks.

It's likely that your FW-1 rule base is quite permissive with your internal hosts, perhaps allowing all traffic that originates within your network to go out to the Internet. The black hole routes ensure that traffic destined for the bogon networks won't pass the firewall, and will, therefore, leave your Internet link and screening router unscathed. Further, because the packet is simply dropped, the performance impact is quite low.

To add a black hole route, use the route command. Here's an example of a black hole route for an RFC1918 network, 10/8, where 8.8.8.1 is the internal (intranet) IP address of our firewall:

```
route add 10.0.0.0 -netmask 255.0.0.0 8.8.8.1
↳-blackhole
```

As a result of this route addition, the firewall will silently discard all packets destined for the 10/8 network. Be careful here, however. Don't add black hole routes for networks that you use internally. We recommend the following black hole routes, which you should add to the end of /etc/init.d/inetinit. Remember to replace 8.8.8.1 with the IP address of the internal interface of your firewall:

```
route add 1.0.0.0 -netmask 255.0.0.0 8.8.8.1 -blackhole
route add 2.0.0.0 -netmask 255.0.0.0 8.8.8.1 -blackhole
route add 10.0.0.0 -netmask 255.0.0.0 8.8.8.1 -blackhole
route add 172.16.0.0 -netmask 255.240.0.0 8.8.8.1 -blackhole
route add 192.168.0.0 -netmask 255.255.0.0 8.8.8.1 -blackhole
route add 192.0.2.0 -netmask 255.255.255.0 8.8.8.1 -blackhole
route add 169.254.0.0 -netmask 255.255.0.0 8.8.8.1 -blackhole
route add 240.0.0.0 -netmask 240.0.0.0 8.8.8.1 -blackhole
```

If necessary (e.g., during an attack), you can add other black hole routes in real time.

Our tuning is now complete. It's wise to reboot at this stage to ensure that all of our changes have been applied and are working as expected.

How's my firewall doing?

Once your finely tuned firewall is in production, it's important to check its health from time to

Listing B: Using netstat to determine our memory usage

```
firewall# netstat -mv
streams allocation:

```

	current	maximum	cumulative total	allocation failures
streams	197	231	606416	0
queues	502	555	1170440	0
mblk	809	7747	1314600	0
dblk_40	409	3444	417488125	0
dblk_72	118	868	63359226	0
dblk_136	50	126	21172406	0
dblk_328	222	3300	5742272	0
dblk_616	0	48	9556749	0
dblk_1096	2	70	68267716	0
dblk_1576	0	60	42263915	0
dblk_1992	0	6	13779515	0
dblk_2664	0	9	82229300	0
dblk_4040	0	2	22651	0
dblk_8136	0	7	73106	0
dblk_12232	0	1	951	0
dblk_esb	0	63	8609885	0
dblk_total	801	8004	732565817	0
linkblk	6	169	8	0
strevent	11	169	7340372	0
syncq	13	56	55709	0
qband	2	127	2	0

350 Kbytes allocated for streams data

time. Fortunately, Solaris provides several commands that will make this possible. You may wish to refer to the CheckPoint FireWall-1 documentation for the FireWall-1 commands.

The first command to use is `netstat -mv`. This command will reveal the memory allocations to the various STREAMS buckets. An example of this output is shown in Listing B.

Watch for any number greater than zero in the allocation failures column of the `dblk` entries. This indicates resource exhaustion, and the possibility that the kernel doesn't have enough memory allocated to the stack. In this case, you may need to increase physical memory.

The output of `netstat -k <INT>`, where `<INT>` is a network interface such as `qfe3`, can be quite helpful as well. This undocumented feature requests interface information from the kernel. Here is an example:

```
root@bilbo# netstat -k le0
le0:
ipackets 9038375 ierrors 415 opackets 1502503
↳oerrors 1 collisions 103594
defer 131451 framing 0 crc 0 oflo 0 uflo 0 missed
↳415 late_collisions 0
retry_error 0 nocarrier 0 inits 67 notmds 0
```



```

↳notbufs 13711 norbufs 420553
nocanput 9387 allocbfail 0 rbytes 1467633274
↳obytes 479889547 multircv 13 multixmt 0
brdcstrcv 133133 brdcstxmt 4285 norcvbuf 9158
↳noxmtbuf 0

```

When reviewing this output, focus on the following fields:

- **collisions.** While our example uses the LANCE Ethernet interface, the HME interface will support a full duplex. You should wire any routing device to a switch that supports a full duplex operation.
- **allocbfail.** The number of times that the driver has been unable to allocate a STREAMS mes-

sage block, e.g. a failure of a call to `allocb`. Verify the issues with the output of `netstat -mv`. It may be necessary to increase the amount of physical RAM in the firewall.

- **nocanput.** The number of times the driver could not send packets upstream due to a full STREAMS queue. The name comes from the `canput()` call. You can resolve this by increasing the `sq_max_size` parameter.
- **notbufs.** This indicates an exhaustion of the transmit buffers in the NIC driver.
- **norbufs.** This indicates an exhaustion of the receive buffers in the NIC driver.

If the `notbufs` or `norbufs` variables are significant, then the driver is overburdened and can't keep pace with the traffic. It will be necessary to decrease the network load on the firewall.

With the output of `netstat -s`, you can actually monitor the amount of traffic the firewall is forwarding. **Listing C** shows an edited sample containing only the IP statistics.

Here we can see that this firewall has forwarded 3,716,846 datagrams out of 3,886,887 datagrams received. There have been few errors of note. However, if errors do occur, examine the TCP and UDP portions of the `netstat -s` output for details regarding the error types.

Listing C: Using `netstat -s` to monitor the firewall traffic

IP	ipForwarding	=	1	ipDefaultTTL	=	255
	ipInReceives	=	3886887	ipInHdrErrors	=	0
	ipInAddrErrors	=	0	ipInCksumErrs	=	0
	ipForwDatagrams	=	3716846	ipForwProhibits	=	89
	ipInUnknownProtos	=	0	ipInDiscards	=	0
	ipInDelivers	=	171327	ipOutRequests	=	86170
	ipOutDiscards	=	0	ipOutNoRoutes	=	0
	ipReasmTimeout	=	60	ipReasmReqds	=	4
	ipReasmOKs	=	4	ipReasmFails	=	0
	ipReasmDuplicates	=	0	ipReasmPartDups	=	0
	ipFragOKs	=	4	ipFragFails	=	0
	ipFragCreates	=	8	ipRoutingDiscards	=	0
	tcpInErrs	=	0	udpNoPorts	=	22
	udpInCksumErrs	=	0	udpInOverflows	=	0
	rawipInOverflows	=	0			

Robust and secure

Providing firewall services is not a trivial task. With a few modifications and careful monitoring, a Solaris box can be configured to provide robust and secure routing for enterprise networks. *

Security: threats and mistakes

by Edgar Danielyan

In the February 2000 article "Defining an Acceptable Use Policy" (on the Web at www.elementkjournals.com/sun/s_sun/0002/sun0024.htm) and the March 2000 article "Securing your networked systems with Solaris 7" (www.elementkjournals.com/sun/s_sun/0003/sun0032.htm), we discussed security issues that we consider both technical and social issues that should be taken into account when developing, implementing and enforcing security policies and procedures. As demonstrated in these articles, if

corporate security is to be a success, it should be both working and enforceable.

In this article, we'll take a close look at the various security threats that face organizations of any size who are connected to the Internet (or to any other public network, for that matter). We'll also examine the common mistakes and omissions made by executives and/or staff that may result in such unpleasant happenings as loss of reputation and customers, bankruptcy proceedings, punitive damages and so on. After identifying

these threats and mistakes, we'll consider various technical and administrative actions that may be taken to either eliminate or minimize the risks associated with them.

Threats

The following are the threats that face almost every network connected to the Internet and have the potential to compromise your network and systems' security. This list is not, by any means, complete, and we doubt that a complete list of security threats may ever be compiled. However, it should give a good understanding of the fact that even the smallest, seemingly harmless security threat may cause a domino effect.

Use of default SNMP community strings—public and private

Too often, even experienced network engineers and administrators install devices on their networks without first making sure that the devices will operate only as expected. Regretfully, that isn't always the case, especially with devices providing SNMP access. Keeping in mind that most, if not all, devices in modern networks have some sort of SNMP access, ranging from primitive and minimal to the full implementation of the protocol. Any device may be a source of SNMP-related security problems.

Such problems may range from a seemingly innocent leak of statistical information about your network interfaces and the model of the device to potentially devastating misuse of read/write access to servers and routers. Unauthorized SNMP access to routers, switches and other network infrastructure devices is especially dangerous, because, in addition to resulting in network downtime, it also may be used to circumvent network-level security systems. Such systems include access lists and network-level firewalls, which, if compromised, may in turn expose systems inside the network that are to be protected from direct access from outside the network.

The most common mistake that results in this sort of security threat is the use of factory settings in network devices. Therefore, you should make sure to change SNMP community strings to something different from the default, off-the-shelf configuration, and set the necessary security options (such as read-only access to SNMP variables).

Buffer overflow problems

The world is imperfect, and so is software (regardless of how much you paid for it!). Most, if not all, software suffers from some type of bug,

with buffer overflow problems being one of the most widespread. Buffer overflow and subsequent unexpected behavior is a dangerous weapon in the hands of an experienced and motivated hacker. Many famous security incidents were possible because of poor software design or configuration.

In many cases, it's possible to use buffer overflow to run arbitrary malicious code with the superuser privileges on the affected machine or network. Buffer overflow problems in some server software, such as IMAP/POP/SMTP mail servers, FTP servers and Web servers, are more dangerous than in others, because in many cases network-level firewalls pass traffic to these ports without much scrutiny. In order to minimize risk, you should install all security-related patches issued by vendors, and, where possible, run servers under a UID other than root's.

Root passwords

Despite the fact that everyone knows that a weak root password is among the worst things to have, many of us still use simple passwords. There isn't much to say in this case: Your firewalls, security notices and door locks are useless if you have weak passwords.

Misconfigured file sharing

Another example of careless attitudes both in Microsoft and UNIX worlds is drives and filesystems being exported without fully assessing security risks and the extent to which access should be allowed to the stored data. Since most firewalls block NFS and SMB traffic at the network boundary, this particular kind of threat is mostly limited to unauthorized access from inside the network (the so-called "insider attack"). So, make sure you fully understand the implications of exporting or mounting a filesystem. A conservative approach to permissions may be appropriate. For example, if the users should only be able to read data from this particular volume/filesystem, but not change it, export it as a read-only volume/filesystem.

Sendmail

Sendmail is a complex and powerful piece of software. Unfortunately, complexity and power bring potential for a misunderstanding, misconfiguration and bugs. The good news, however, is that it's constantly under development, with bug fixes and security updates being released periodically. Therefore, the soundest advice would be to keep an eye on news from the Sendmail Consortium

(www.sendmail.org) and always install the latest release-level versions and patches. For large sites, having centralized mail storage and delivery systems will help you retain overall control over sendmail behavior.

Use of alpha or beta stage software

All of us like to experiment with new software or software with cool new features. However, a production system isn't the place for experiments. Therefore, try to avoid at all costs the use of alpha or beta software on your production systems—no matter how great the temptation. Software in alpha and beta stages of development is bound to have at least a few bugs, and we have enough of them in production software!

RPC issues

Software that uses a Remote Procedure Call (RPC) interface is both complex and important to the operation of the system. Try to run only those RPC services that you really need and use, and disable all others. Use security features (GSS, Diffie Hellman, etc.) where available.

Bad CGI scripts

With the ever-expanding use of the World Wide Web and dynamic Web sites, use of CGI scripts and Web programming languages will only increase with time. Today, many Web sites are completely dynamic, with Web pages being generated on the fly from the information stored in a database and displayed in a customized manner tailored to a particular user. All of this comes at a cost, in particular at the cost of security. Some CGI-related security threats, such as variables passing between an HTML form and a CGI script, have a higher potential for misuse. It's difficult to recommend a universal solution to this threat, so the common-sense approach would be the best solution: Test your software before installing it on a production system, and continuously monitor your server's logs.

BIND

Bugs in the Berkeley Internet Name Domain (BIND) server software may play their roles in an attack, especially keeping in mind that many services extensively use the Domain Name System. Given the complex nature of the system in general, and the implementation in particular, the most realistic recommendation would be to always keep your BIND servers up to date with the latest release-level versions, and apply patches issued by the Internet Software Consortium (www.isc.org).

Executive mistakes

While most executives understand that security is vital for the well being of the company, very few completely understand the complex relationships between technical and administrative factors that influence security decisions and efficiency. Let's take a look at some mistakes often made by the executive-level staff of a company.

Pretending everything is fine

While it may sound too strange to be true, some executives like to think that security problems will leave them alone if they don't pay attention to them. We see no logic in this approach, and the only thing we can say to these ladies and gentlemen is that sooner or later they will realize how wrong they were—but it may be too late.

Management shortsightedness

Increasingly under pressure to deliver results in this fast-paced world of the 21st century, managers often resort to making decisions in favor of quick fixes, without considering the implications of such quick fixes. In many cases, the results of a problem are softened, rather than eliminating or minimizing the cause. These managers should remember that only long-term planning may help in the long-term development and survival of the company—and, after all, these one-day fixes are only good ... for one day.

"We have a firewall!"

This is the reply we hear most often from executives when asked what security systems and procedures they have in place. What most of them fail to understand is that firewalls are only one piece of a jigsaw puzzle called *enterprise security*, and no firewall can protect you from uneducated employees and experienced and motivated hackers. Firewalls are necessary but not enough to provide a complete security framework. They must be used in concert with other security measures, both technical and administrative.

Consistent and complete

These are two words you should be able to use when describing your security procedures. If a security policy isn't consistent or complete, there will always be a way to find a workaround. There's no point in having a firewall if your firewall's root password is *password*! There's no point in using strong passwords if you connect using telnet. There are countless more examples of inconsistency and incompleteness that may contribute to the weakening of security.

Failure to understand

Executives need to understand the relationship between their organization's computer systems and business mission. There's no such thing as useless information; any information in the hands of an experienced hacker may be used to obtain more information. Of course, a bank's PIN codes are more important than a country club's records, but both have a price tag. The challenge is to have appropriate security measures in place depending on the nature, value and importance of the information.

People are everything

Due to the constant shortage of skilled information technology personnel, some organizations are forced to employ untrained and inexperienced staff. While there's no easy solution to this problem, care must be taken to ensure that the security of your information technology infrastructure isn't in the hands of a person with little or no experience in security.

User mistakes

In many cases, end users of the network or a computer system unknowingly increase or contribute to the level of security threats. You should properly train users and constantly remind them of the ever-present security issues.

Modem connected to a PC inside a secure LAN

This is one of the most widespread and dangerous mistakes. A user connects a modem to his desktop PC at the office in order to work after hours—an intention probably applauded by the company's management. However, security implications of such a setup are paramount, and effectively reduce the return on investment in a company's firewall and intrusion detection systems to null. The solution: State in your acceptable use policy that no modems of any kind can be connected to any devices in the company's network without express written permission of the network administrator, and enforce this policy without exceptions.

Backups

Everyone understands that backing up important data and software is a must, but many times this is an afterthought. Good backups are an integral part of a security policy. Make backups and test them—you never know when you'll need them.

New screensaver

Such things as screensavers, games and so on, may contain malicious code, and when down-

loaded and executed inside the network may be of considerable danger, since most firewalls don't expect an attack from inside the network. Use both technical and administrative procedures to inform your staff that it's prohibited to download any software from outside the trusted network without the network administrator's prior written consent.

Active documents

Your acceptable use policy should prohibit the opening of attachments sent by regular Internet email without authentication or digital signatures. Such software as Microsoft Office and others provide macro-programming facilities that may be grossly misused. In large organizations, internal and Internet email systems should be separate.

Administrator mistakes

We all make mistakes. The point is to identify and fix them before they are discovered by hackers and intruders.

Absence of security incident response plan

Ignoring a problem isn't a good way to solve it. Therefore, have a plan of action for situations when the breach of security has already happened. Educate your staff and coordinate your actions with the management of the organization to minimize losses and to stop unauthorized activity.

Viruses

While not an issue in UNIX-only installations, viruses are a real problem in the Microsoft PC world, as all of us know. Even if you have only a pair of PCs, use anti-virus software and always keep it updated.

Misconfiguration

Misconfiguration of a single firewall will place the entire network under risk of intrusion and unauthorized access. Have testing and quality control procedures in place to ensure that the configuration of live production systems actually works. There's a big difference between a system that *should* work and a system that *does* work.

Unnecessary services

Avoid running services and servers that you don't use. If there's no service offered, it can't be abused. The first (but not the only) thing to be disabled are the TCP and UDP small servers, such as chargen, echo, discard daytime, and so on.

Enterprise-level backups

Even if users of individual systems and PCs back them up themselves, make double backups of the most important servers, such as database, Web and DNS servers. Test the backups regularly. Don't change anything in your backup procedures without first checking whether the new way of doing things really works.

Authenticate everything

Don't take anyone's word, especially conveyed by unauthenticated methods, such as regular email. Make sure that sensitive information isn't sent by email, even inside the enterprise. Take care in what information you give over the phone. Don't take instructions from management without being absolutely sure they are genuine and really come from the stated source.

Eavesdropping

Always remember the risk of eavesdropping. Use SSH or other ways of encrypting and authenticating connections. Keep in mind that the entire chain of connections should be secured; not just part of it. SSH would be of no benefit if you telnet to your ISP from home and then use SSH to connect to your enterprise's server.

Monitoring

There's no such thing as plug-and-play security. Constant and scrupulous monitoring of all important parts of the network is essential for early detection of problems. The sooner you're aware of a problem, the sooner you'll be able to solve it and minimize the losses. Every device on your network can be a potential security threat. Don't take anything for granted. *

OCR for Solaris

by Clayton E. Crooks II

The paperless office that was supposed to materialize in the last decade has yet to become reality. The idea is certainly a sound one, and although technology continues to become a greater part of our daily lives, we're really not any closer to the goal than we were a decade ago. That's not to say that converting paper documents to a digital format is useless. Digital documents provide far superior qualities in several areas, the most notable of which is the ability to search through hundreds of documents in seconds instead of the hours it would take to do by fingering through stacks of paper. Another advantage is the ability to edit documents quickly and easily. And digital formats even save a few trees.

The ability to turn printed materials into text information is the responsibility of scanners and Optical Character Recognition (OCR) software. Without going into enormous detail, OCR software converts pages of graphics files into text. In order to accomplish the task, the software compares each character with a selection in various fonts stored in a database. The software makes an accurate calculation of what the character is and then adds it to a new text document.

UNIX software

Vividata released the first version of OCR Shop, which is based around technology licensed from Caere Corporation in 1997. This product has

quickly evolved and is now the easiest and fastest OCR package available for UNIX. It's worth mentioning at this time that the Caere OmniPage engine that this product is based around has won a variety of awards on the PC and Mac platforms.

Installation

The software installation went without a hitch and uses a Motif-based graphical user interface (GUI) tool. You must start the Installer from a console or terminal window within an X-based window manager. According to the documentation, you may be required to interact with terminal messages, so don't launch the installer from a file manager utility such as dtfile.

Two versions are available, and offer slightly different installation instructions. A downloadable version is available from the Vividata Web site (www.vividata.com) that requires extraction of the installation files and then running the installer. A CD is also available from Vividata, which provides a slightly easier installation process, although the difference is negligible. Both solutions offer easy installation, but be careful to read the documentation for specific actions needed with certain versions of UNIX.

Potential OCR problems

There are a variety of potential problems that are encountered when using OCR on a document. For

instance, many characters look the same to the software. For instance, an OCR engine may be forced to choose between a lowercase *l* and the number *1*. Another potential problem is the production of old documents that may have remnants of years of use and storage. These may appear as speckles or wrinkles on a sheet of paper, and when scanned, they often can confuse the OCR engine into thinking a character is something different or that there's a character where none actually exists.

The OmniPage engine

It's the job of the OCR software to handle all of these potential pitfalls, and we previously mentioned that the OCR Shop software is based around the Caere OmniPage engine. The OmniPage OCR engine provides a variety of enhanced functionality that helps the OCR Shop software deal with many of the mentioned difficulties. The first, and maybe most important, of the improved features is the use of a technology called AnyFont that allows the software to recognize an entire page of text using a series of steps they call experts.

The first expert evaluates the image and decides if it's a character for which the expert is responsible. If it's not certain, it passes the character on to the next expert. This continues until the appropriate expert recognizes each character. As a result, unlike other OCR engines, no probabilities or guesses are used. This not only improves speed, but also enhances accuracy.

Another interesting idea is the use of a technique called *computerized compound neural system*, which recognizes incomplete characters much like our brain recognizes letters when some characters are fragmented. This system consists of several neural networks that are made up of rows and columns containing software-stimulated neurons. Each neuron weighs evidence provided by the image, as well as information provided by other neurons within the network, to compute the probability of specific possible characters.

The final engine features we'll look at are Language Analyst and Trigram Analysis. Together, they offer advanced features that allow the software to make changes based on a series of expectations. For instance, when confronted with the English language, the engine will generally expect

the letter *u* to follow the letter *q* and knows that *qui* is a more likely letter combination than *qul*.

The OmniPage enhancements are individually quite effective. When they are all combined, they provide an excellent solution for even the most difficult of scanning situations.

Language support

OCR Shop can recognize text in 14 languages, including French, German, Spanish and English, which is supported for both the United Kingdom and United States. There are also a variety of solutions for specific industries, including medical and legal, and there's also the ability to provide your own custom dictionaries.

Input and output formats

While the engine itself could be outstanding, an OCR package that doesn't provide a variety of input and output formats would be nearly useless. Luckily, the OCR Shop software provides a tremendous number of each.

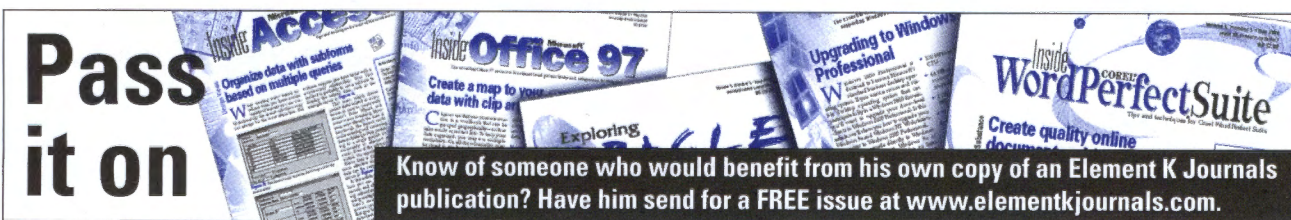
Input formats include CCITT Group 3 and 4, Encapsulated PDF, EPSI, GIF, JPEG, LZW, PBM, PNG, SGI RGB, Sun Raster, X11 Bitmap, XWD and TIFF. The output formats are equally impressive and include ANSI (standard, stripped), Microsoft Word, Microsoft Excel, Lotus, dBase, WordPerfect, Word Star and HTML to name a few of the more popular formats.

A total OCR solution

If you're looking for a full-feature application for OCR, Vividata may have just what you need. You can fully evaluate OCR Shop by downloading the demo version from the Vividata Web site.

It's a complete package that's good right out of the box and will quickly become even more reliable after you customize it for your particular needs. The demo doesn't allow you to save the exported text, but it does allow you to experiment with the software to see the speed and accuracy at which it performs. It's definitely worth the download and installation time.

OCR Shop is available in personal, non-profit and corporate versions. The prices start at \$99 for personal, and move up to \$745 for non-profit and \$1,495 for corporate. *



Pass it on

Know of someone who would benefit from his own copy of an Element K Journals publication? Have him send for a FREE issue at www.elementkjournals.com.

Exploring REGEXs with grep

by Don Kuenz

Many languages and applications allow you to filter text by using Regular Expressions (REGEXs). Languages, such as C, allow you to embed REGEXs into your software. Applications allow you to use REGEXs within filter files. The three tools, fgrep, grep and particularly egrep, provide an easy way to explore REGEXs.

In this article, we'll show you how to use each of the three grep tools. We'll start off explaining fgrep, the simplest of the three; then we'll talk about grep. Finally, we'll show you how to use egrep to develop and debug filters that use REGEXs.

fgrep

All three of the grep tools share a similar syntax. Generally, you enter the tool name, followed by

two arguments. The general syntax used by each tool is as follows:

```
fgrep 'searchPattern' fileList
grep 'searchPattern' fileList
egrep 'searchPattern' fileList
```

The first argument contains a search pattern, while the second argument contains a file list. Depending upon the tool, the search pattern may or may not allow regular expressions that use metacharacters such as the asterisk (*) and dollar sign (\$).

All tools search the file list and display any lines that contain the search pattern. Regardless of the grep tool that you use, you should always enclose the search pattern within single quotes to prevent your shell from incorrectly interpreting the search pattern.

Let's take a closer look at fgrep. The f in fgrep stands for *fast*. fgrep trades off versatility for speed. It searches files faster than the other two tools because it interprets the search pattern literally. However, fgrep doesn't interpret any metacharacters contained in the search pattern.

Listing A shows the contents of a file named *junk* that we use as input to each of the tools. **Listing B** shows the results produced by an fgrep '.' *junk* command. Notice that fgrep interprets '.' literally and displays every line that contains a period.

grep

The grep tool improves upon fgrep by allowing metacharacters in its search pattern. **Table A** shows some of the metacharacters recognized by both grep and egrep.

Listing A: *The contents of a file named junk that we use as input to fgrep, grep and egrep*

```
From: fohv@jhnrvpqfulo.gov
Subject: yo ehkgbvl exnko ntpc oyszki
➔crnlscw ekr iflep scv cleq
Newsgroups: alt.test
Message-ID: <8F4873BAE60.DQHHAYNJ@PMLLAFHHWFX.NET>
Date: Sun, 18 Jun 2000 00:46:06 GMT
Organization: Newsserver BayCIX GmbH
Lines: 48
NNTP-Posting-Host: 195.226.187.146
X-Trace: ninja.muc.baycix.de 961189046
➔30553 195.226.187.146
X-Complaints-To: usenet@news.muc.baycix.de
NNTP-Posting-Date: 16 Jun 2000 20:57:26 GMT
Path: news.maxwell.syr.edu!newsfeed.cwix.com!wn3feed
Xref: apollo alt.test:25757
X-Mozilla-Status: 8000
X-Mozilla-Status2: 00000000
```

```
Y frf lo ckdc mf ti bb fb spnl oe eeiew
lbpvt zaent kare oie y ifiql beaoa kv
empfmb johnhv uf ictllw sefptu myfspz
➔zfoa vqskf vowamme a lbcae
fbtdktykr set aulkolrt lsntotb mfr pehpfhmmi ttau fiilr
kuxdw flcr tbfir y alr fukiv o wbyko eyd ym a edrne
grfsy a ooin sur uiny lpayv ipkfqasom
➔epsrnbjgj xfifphj tbtesme le. ziiem
➔iyyptes dhbr fat lzk jv
bkeng vfrkqt ujj bv vaeme ouelg
hphr i bryek ojxdu rie kezim
hke i tcj xejn lbb
```

Listing B: *The results of the fgrep command*

```
$ fgrep '.' junk
From: fohv@jhnrvpqfulo.gov
Newsgroups: alt.test
Message-ID: <8F4873BAE60.DQHHAYNJ@PMLLAFHHWFX.NET>
NNTP-Posting-Host: 195.226.187.146
X-Trace: ninja.muc.baycix.de 961189046 30553
➔195.226.187.146
X-Complaints-To: usenet@news.muc.baycix.de
Path: news.maxwell.syr.edu!newsfeed.cwix.com!wn3feed
Xref: apollo alt.test:25757
grfsy a ooin sur uiny lpayv ipkfqasom
➔epsrnbjgj xfifphj tbtesme le.
```


Listing C shows the results produced by a `grep '.'` junk command. Notice that `grep` displays almost every line from the junk file. This occurs because `grep` treats the period in the search pattern as a metacharacter. **Table A** shows that the period metacharacter matches any single character. So `grep` displays all of the lines in junk except the single blank line about halfway down in **Listing A**. `grep` ignores the blank line because it contains no characters.

You can make `grep` display the same results that `fgrep` did in **Listing B** by disabling the special meaning of the period metacharacter by prefixing it with a backslash. A `grep '\.'` junk command displays the same lines shown in **Listing B**.

egrep

Most users find that `fgrep` and `grep` meet most of their needs. On the other hand, administrators charged with maintaining filtering configuration files must understand how to use `egrep`, the most advanced of the three greps.

In addition to understanding the metacharacters shown in **Table A**, `egrep` also understands pipe (`|`), and opening and closing parentheses (`()`), and a few more metacharacters. The pipe ORs together patterns, while the parentheses allow you to establish precedence. This enables you to create the powerful REGEX expressions that today's administrator needs to perform real-world filtering. You can create and debug a REGEX using `egrep`. After you prove that it correctly filters, you can usually paste the resulting search pattern, the characters between the single quotes, into a configuration file like `newsbot.conf` or `suckkillfile`.

The contents of junk, shown in **Listing A** came from an actual news article. Usenet abusers post such articles in an attempt to render a news group unusable. Let's see how we can use `egrep` to perfect a REGEX, which detects such articles. After we create the REGEX, we can place it into either `newsbot.conf` or `suckkillfile` to keep such articles out of our local news server.

We can create a search pattern based upon the Message-ID to detect such articles. **Listing D** shows such a pattern. Keep in mind that we created the search pattern shown in **Listing D** piece by piece.

Before you start creating a REGEX, you need to find a pattern. You can master `egrep` by continuously using it. Unfortunately, you'll find that deriving a pattern is more of an art than a science.

If you examine junk's Message-ID closely, you'll see that it contains the following tokens in order:

Table A: Metacharacters used in search patterns by both `grep` and `egrep`

Metacharacter	What it matches
\c	Disables special meaning of metacharacter <i>c</i>
^	Begins a line
\$	Ends a line
.	Indicates a single character
*	Indicates zero or more occurrences of the previous character
[]	Indicates any of the characters between the brackets
[^]	Indicates any character not between the brackets

Listing C: Results of a `grep` command

```
$ grep '.' junk
From: fohv@jhnrvpqufulo.gov
Subject: yo ehkgbvl exnko ntpc oyszki
↳crnlscw ekr iflep scv cleq
Newsgroups: alt.test
Message-ID: <8F4873BAE60.DQHHAYNJ@PMLLAFHHWFX.NET>
Date: Sun, 18 Jun 2000 00:46:06 GMT
Organization: Newsserver BayCIX GmbH
Lines: 48
NNTP-Posting-Host: 195.226.187.146
X-Trace: ninja.muc.baycix.de 961189046
↳30553 195.226.187.146
X-Complaints-To: usenet@news.muc.baycix.de
NNTP-Posting-Date: 16 Jun 2000 20:57:26 GMT
Path: news.maxwell.syr.edu!newsfeed.cwix.com!wn3feed
Xref: apollo alt.test:25757
X-Mozilla-Status: 8000
X-Mozilla-Status2: 00000000
Y frf lo ckdc mf ti bb fb spnl oe eeiew
lbpvt zaent kare oie y ifiql beaoa kv
empmb johnhv uf ictllw sefptu myfspz
↳zfoa vgskf vowamme a lbcae
fbtdktykr set aulkolrt lsntotb mfr
↳pehpfhmmi ttau ffilr
kuxdw flcr tbfr y alr fukiv o wbyko
↳eyd ym a edrne
grfsy a ooin sur uiny lpayv ipkfqasom
↳epsrnbjgj xfifphj tbtesme le.
```

Listing D: Results of an `egrep` command

```
$ egrep 'Message-ID: <([0-9A-F]*)[A-F]([0-9A-F]*)
↳\.[(A-Z)*]@[A-Z]*\.(COM|EDU|MIL|NET|GOV|ORG)>.' junk
Message-ID: <8F4873BAE60.DQHHAYNJ@PMLLAFHHWFX.NET>
```



```
Message-ID: <
Uppercase hexadecimal number
.
uppercase alphanumeric string
@
uppercase alphabetic string
.
COM or EDU or MIL or NET or GOV or ORG
>
```

Although this article only shows a Message-ID ending that contains NET, practically speaking, usenet abusers also use COM, EDU, MIL, GOV and ORG. The second to last token also allows you to see how to use an OR in a REGEX.

Let's start building our REGEX by putting all of the literals together. When we do that, we wind up with the following:

```
Message-ID: <.*\..*@\..*>
```

Notice that we prefix some of the periods with a backslash to disable their special meaning as metacharacters. The other periods and asterisks still function as metacharacters, and **Table A** tells us that they will match one or more occurrences of any character. Although this search pattern identifies junk, it also creates a lot of false hits.

Let's make our search pattern more accurate by replacing the final `.*` with `(COM|EDU|MIL|NET|GOV`

|ORG). The parentheses, combined with the pipe metacharacter tells egrep that it must match on one of the three character strings. That eliminates a few more false hits, but we still need to do better.

To make our search pattern even better, we can specify an alphabetic string with `([A-Z])` and an uppercase alphanumeric string with `([A-Z0-9])`. That improves our REGEX considerably, but it still generates false hits.

Let's tackle the hexadecimal number. You might be tempted to specify the hexadecimal number using `[A-F0-9]`, but that doesn't work, because it matches strings that contain only digits, which will generate false hits. Now, we need to use some of the art mentioned earlier.

The odds of a 10-byte hexadecimal number containing all digits are rather long. We gamble that our hexadecimal number contains at least one letter. That allows us to maximize positive hits and minimize false hits. We use `([0-9A-F]*)([A-F])([0-9A-F]*)` to specify a hexadecimal number that contains at least one letter.

Conclusion

In this article, we've shown you how to use the three grep tools that Sun includes with Solaris—fgrep, grep and egrep. These tools provide a powerful mechanism for manipulating text. Each tool has its own strength, so make sure you choose the right one for the job. *

Inside Solaris 2000 index

This index is arranged by month, listing the topics covered in *Inside Solaris* during 2000. Element K Journals provides this index to make the issues you received in 2000 more useful as a problem-solving resource and to help you lo-

cate articles of interest in issues missing from your collection. You can order back issues from the last six months by calling Customer Relations at (800) 223-8720. All issues are also available online at www.elementkjournals.com/sun.

January

- A productive K desktop environment—KDE
- Understanding the /proc file system
- Developing Solaris knowledge bases
- Virtual memory and priority paging with Solaris 7

February

- Benchmarking and performance measurement
- Securing systems with ASET

- /etc/system

- Defining an Acceptable Use Policy
- Run Linux applications on your Solaris with Ixrun
- ICQ on Solaris

March

- Using Apache as a proxy server
- Securing your networked systems with Solaris 7
- Introducing Message Digest algorithm, version 5

Packaging in Solaris
 Solaris moves toward open source
 Code coverage and profiling with Tcov
 Making cron jobs quiet
 It's not a bug, it's a feature
 Network management for free
 Solaris Device Configuration Assistant
 Why is this machine so slow?
 Dual booting Solaris and Linux

April

Turn a Solaris box into a packet-filtering firewall
 Secure intranet file transfers
 Inprise releases JBuilder 3 Solaris Edition
 Easy device driver development
 The Common Desktop Environment
 Installing Solaris x86
 Recordable CD-ROMs for Solaris
 Solaris Device Configuration Assistant
 Turning off diagnostic mode
 Adding a second IP address without another network card
 Configuring Solaris to recognize a second network interface card
 Boot using the 32-bit kernel

May

Roll out your own high availability
 Synchronizing computer clocks
 My file system is full! Now what?

About our contributors

Clayton E. Crooks 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.

Edgar Danielyan is currently self-employed. His list of qualifications include Cisco Certified Network Associate, diploma in company law from the British Institute of Legal Executives, and certified paralegal from the University of Southern Colorado. He has been working as a network administrator and manager of a top-level domain of Armenia. He's also worked for the United Nations, the ministry of defense, a national telco, a bank, and has been a partner in a law firm. He speaks four languages, likes good tea, and is a member of ACM, IEEE CS, USENIX, CIPS, ISOC and IPG, to name a few. He can be reached at edd@danielyan.com.

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.

Rob Thomas is a systems, network, and security architect with the professional services division of a large telco. He can be reached at robt@cymru.com, or visit his home page at www.enteract.com/~robt.

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

EditorGarrett Suhm

Assistant EditorJill Suhm
 Managing EditorMichelle Rogers
 Assistant Managing Editor.....Dianne Galloway
 Copy Editors.....Rachel Krayer

Contributing EditorsGlenna Lechner
 Clayton E. Crooks II
 Edgar Danielyan
 Don Kuenz
 Rob Thomas

Graphic Designer.....Rachel J. King
 Cover and Content Design.....Melissa Ribaldo

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

© 2000 Element K Content LLC. All rights reserved. Reproduction in whole or in part in any form or medium without express written permission of Element K Content LLC is prohibited. Element K is a service mark of Element K LLC. *Inside Solaris* is an independently produced publication of Element K Journals. Element K Journals reserves the right, with respect to submissions, to revise, republish and authorize its readers to use the tips submitted for personal and commercial use. For reprint information, please contact Copyright Clearing Center, (978) 750-8400.

Inside Solaris is a trademark of Element K Journals. Sun, Sun Microsystems, the Sun logo, SunSoft, the SunSoft logo, Solaris, SunOS, Suninstall, OpenBoot, OpenWindows, DeskSet, ONC and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. Other brand and product names are trademarks or registered trademarks of their respective companies.

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.
 GST# 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

- Interbase goes open source
- Security and cryptography

USPS ARMIN PS1 881 APPROVED POLY

Provide the required security for your systems and data

Trojan horse software and denial of service

Solstice DiskSuite 4.x—tips and pitfalls, part 1

Creating multiple subdirectory levels at once with `mkdir -p`

June

Solstice DiskSuite 4.x—tips and pitfalls, part 2

Build your own IDS with Logsurfer

Conveniently administrating remote servers

Take control of your man files

July

Accelerating development with Enterprise Java Beans

Sun and Palm team up for remote wireless access

Samba password encryption

Network security with Kerberos

Using external SCSI devices with Solaris x86 on a PC system

Tracking `.rhosts` files

Solaris CD recording

August

PostgreSQL—a free SQL database for Solaris

MetaFrame for Solaris—a thin-client alternative

Understanding run levels and `/etc/rc2.d`

Converting PDF files to HTML files with `pdftohtml`

Configuring BIND 8

Easily create multiple levels of directories

Create a one-drive partition

September

Porting applications between NT and UNIX

Squid basics

PERIODICALS MAIL

2096



*****3-DIGIT 480

C: 7661905 00002096 03/01

RUDOLPH LIEDTKE

RJL SYSTEMS

33955 HARPER AVE

CLINTON TOWNSHIP MI 48035-4218

28

61

Sniff your own networks with `tcpdump`

Distributed computing with CORBA and Java

Send output and errors where you want

Use type to find commands and learn more about them

Why making copies of UNIX CDs doesn't work

October

Email for a small business

Hooking into news

Taking advantage of ToolTalk

Space—the final frontier

Replicate filesystems and directory hierarchies with `rsync`

Creating your own tunnel

November

Making space with partition switching

Visualizing CPU activity

Determining the number of processors

Windows emulation on Solaris

Newsbot cleans up

Configuring BIND 8: Part 2

Cross-platform ASP

Renaming your Sun computer

Virtual interfaces on Solaris

Looking out for `setuid` programs

December

Supercharging third-party libraries with memory-mapped files

Securing BIND on Solaris

Auditing Solaris security with CLI

Using `truss` to track processes

Low-hassle news pulls *

Low-cost Solaris

If you're looking for an inexpensive copy of Solaris for personal use, you'll find that you can get a copy and only pay only media, shipping and handling costs. If you're an educational user, you'll find more information at www.sun.com/edu/solaris. For everyone else, go to www.sun.com/solaris/freesolaris.html for more information.

The cost for Solaris 8 is \$75 for the package that includes Solaris software such as StarOffice 5.1, iPlanet, Netscape, Oracle8i, AnswerBook documentation and a CD full of open-source software.