AUUG Membership and General Correspondence

The AUUG Secretary
PO Box 366
Kensington NSW 2033
Telephone: 02 8824 9511
or 1800 625 655 (Toll-Free)
Facsimile: 02 8824 9522
Email: auug@auug.org.au

AUUG Management Committee
Email: auugexec@auug.org.au

David Purdue
iPlanet e-commerce solutions
Level 5
276 St Kilda Road
Melbourne, Victoria, 3004
<David.Purdue@auug.org.au>

Vice-President
Michael Paddon
<Michael.Paddon@auug.org.au>

Secretary
Greg Lehey
IBM Australia
PO Box 460
Echunga, SA, 5153
<Greg.Lehey@auug.org.au>

Treasurer
Luigi Cantoni
Objective Management Pty Ltd
PO Box 51
North Perth WA 6906
<Luigi.Cantoni@auug.org.au>

Committee Members

Warren Toomey <warren.toomey@auug.org.au>
Sarah Bolderoff <sarah.bolderoff@auug.org.au>
Peter Gray <peter.gray@auug.org.au>
Conrad Parker <conrad.parker@auug.org.au>
Malcolm Caldwell <malcolm.caldwell@auug.org.au>

AUUG Business Manager
Elizabeth Carroll
PO Box 366
Kensington NSW 2033
<busmgr@auug.org.au>

Editorial

Con Zymaris
auugn@auug.org.au

Welcome to another edition of AUUG's official journal. As you know, AUUG is a user-group organisation, whose focus has been in the communal interest of all things Unix®. Included in this are knowledge topics which either began with Unix or were heralded loudest by Unix, but have now become totally pervasive in our computing landscape: TCP/IP, vendor-independence, sockets programming, open systems, heterogeneous intercommunications, standards, network security, C, C++, Java, Perl, Python, the Internet, remote-execution of applications, the Web, open source...

As each of these technologies, methodologies or philosophies grew, it followed a certain, reasonably reproducable path. From slow beginnings, uptake by early practitioners, a flurry of interest and activity, and eventual adoption by ***hoi poloi.*** In short, we have seen many of these platforms change from torrential free-flowing proto-rivers bursting with possibilities, to slowly meandering, but graceful and still self-assured, mature deltas of general practice. The earlier we take our snapshot of this flow of adoption for one of the technologies, the greater the rate-of-change of information material, research, implementation suggestions, case histories etc. It should come as no surprise then that some topics are presently generating the bulk of the material which cross your editor's desk, for possible inclusion in AUUGN. The livewire topics of the current day are open source platforms and technologies, web and distributed development and network security.

Still, having come from an age when the hot topics where BSD 4.1 on Vaxen, and knowing how much this section of our industry changes year-to-year, I feel it important to keep a diversity of topic and ***materiel*** in each issue of AUUGN. To that end, I went looking for the kinds of articles that ***used*** to be in AUUGN. As those of you who have memories of this journal dating back 10 or so years ago will atest, articles on ***vendor*** Unix were all the rage. The vendors (Sun, HP, IBM, Digital, ***my gawd*** Apollo...) did a lot to assist their burgeoning communities in their uptake and practice of Unix.

So, here are my results. I talked (and am still talking) with Paul McKeon, Marketing PR Specialist, IBM. I talked with Matt Ruetz, Solaris Outbound Marketing Sun Microsystems Inc., who runs Sun's BigAdmin knowledge repository and finally with Lawrence A. Waskom of MarketLINK (formerly of ***RS/6000 Results*** magazine.) In general, the discussions have not lead to actual material for AUUGN, but I'm hopeful. In the meantime, you know what to do if you have contacts in the right places within Unix vendor organisations, right?

Cheers,
Con

Contribution Deadlines for AUUGN in 2002

Volume 23 • Number 1 – February 2002: **January 15th, 2002**

Volume 23 • Number 2 – May 2002: **April 15th, 2001**

Volume 23 • Number 3 – August 2002: **July 15th, 2001**

AUUGN Editorial Committee

The AUUGN Editorial Committee can be reached by sending email to:
auugn@auug.org.au

Or to the following address:
AUUGN Editor
PO Box 366
Kensington NSW 2033

Editor:
Con Zymaris

Sub−Editors:
Frank Crawford

Public Relations and Marketing:
Elizabeth Carroll

AUUGN Submission Guidelines

Submission guidelines for AUUGN contributions can be obtained from the AUUG World Wide Web site at:

www.auug.org.au

Alternately, send email to the above correspondence address, requesting a copy.

AUUGN Back Issues

A variety of back issues of AUUGN are still available. For price and availability please contact the AUUG Secretariat, or write to:
AUUG Inc.
Back Issues Department
PO Box 366
Kensington NSW 2033

Conference Proceedings
A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available. Contact the AUUG Secretariat for details.

Mailing Lists

Enquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat.

Disclaimer

Opinions expressed by the authors and reviewers are not necessarily those of AUUG Inc., its Journal, or its editorial committee.

Copyright Information

President's Column

David Purdue
David.Purdue@auug.org.au

On the 27th of November, 2001, ISOC-AU and AUUG sent a joint submission to Judge J. Frederick Motz, who is considering remedies in the Microsoft anti-trust case. Here is the text of that submission.

Hon. Judge J. Frederick Motz
H.S. District Court for the District of Maryland
Fax #: (410) 962-7574

Dear Sir,

The Internet Society of Australia (ISOC-AU) and the Australian Unix and Open Systems Users Group (AUUG Inc.) are two organsations that combined represent the interests of over 20,000 Internet and computer users and consumers in Australia (see http://www.isoc-au.org.au and http://www.auug.org.au). In addition, ISOC-AU is a chapter of the Internet Society (http://www.isoc.org) which provides leadership in addressing issues that confront the future of the Internet, and is the organization home for the groups responsible for Internet infrastructure standards including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB).

It has recently come to our attention that Microsoft has proposed a deal to settle a number of private antitrust cases in relation to alleged anti-competitive behaviour of the company. It is our understanding that this deal includes a proposal donate computer hardware, software and support to 14,000 poor school districts throughout the United States, and that under the proposed settlement a substantial part of the value provided to schools would be in the form of Microsoft software.

ISOC-AU and AUUG applaud the sentiment expressed in the Microsoft proposal and fully support any initiatives that are designed to enhance computer and Internet literacy and reduce the ever widening divide between the information rich and the information poor. However, we have a number of concerns in relation to the specific details of the Microsoft settlement. These include:

1. The consequences of the actions perpetrated by Microsoft extend far wider than the borders of the United States of America. We believe that at least 50% of any settlement should be made available to developing and third world countries of the world who are less able to defend themselves against the type of practices that have been demonstrated by Microsoft, but should nevertheless be entitled to any remedy that is intended to redress these practices.

2. Our concern that by including the Microsoft Windows operating system as part of the settlement, Microsoft will be effectively extending and enhancing its monopolistic practices. Further, it is our opinion that such a settlement would derive significant advantage for Microsoft and that this is against the spirit and intent of such a penalty.

In relation to this second item, ISOC-AU and AUUG would support an approach in which Microsoft provides financial assistance to purchase generic computer hardware but leave the recipients free to choose the appropriate software to operate the computer systems.

ISOC-AU and AUUG would appreciate your consideration of these issues when making your final decision whether or not to accept the settlement.

Yours sincerely,

Greg Watson, President ISOC-AU
David Purdue, President AUUG

/var/spool/mail/auugn

Editor: <auugn@auug.org.au>

What follows are ***none*** of the regular AUUG-related email exchanges, due to the fact that nonesuch have crossed your editor's desk in recent times! Instead, I've trolled many a site looking for the kind of mail I would hope to find hitting my intray, and also populating the auug-talk mailing list. Speaking of which, If you want to contribute to the list, mail majordomo@tip.net.au with:

```
subscribe talk Your Name <your@email.com.au>
```

Date: Tue, 04 Mar 1997 17:53:16 -0800
From: Todd E Van Hoosear
<vanhoose@cl-next4.cl.msu.edu>

Subject: Hints and Tips to get the best performance from your Sysadmin

Hints and Tips to get the best performance from your Sysadmin

1. Do not ask your sysadmin "did you get my mail?". Your sysadmin receives more mail in an hour than you do in a week, and may well have already read and forgotten your mail. If he hasn't answered it could be that he has more important things to do, like restoring the passwd file on the main server.
2. Do not page the sysadmin at 1am to ask him simple shell programming questions. Your sysadmin has made the wonderful and enlightening set of UNIX man pages available to you to answer just that kind of question.
3. When in doubt, assume that it's your fault. It probably is.
4. If the networks' down and your sysadmin is laboring feverishly in the machine room, please do not pound on the machine room door to tell him that the network's down. He already knows.
5. Overly-general questions like "what's wrong with my computer?" or "what did you do the network?" do little except annoy the sysadmin and make him quiz you to find the actual symptoms that you are experiencing.
6. Accusing your sysadmin of favoritisim ("you won't fix my problem because you like the other engineers better") is infantile and ridiculous. Your sysadmin holds all users in equal disdain and is ignoring your problem because he has more important problems to deal with.
7. Do not, under any circumstances, walk into your sysadmin's cubicle and announce "I have no problem, I just wanted to tell you what a wonderful job you're doing" unless you want your sysadmin to drop dead from shock.

http://www.lne.com/ericm/sysadmin.html

From: tai@bbo.memphis.edu (Tai)
Subject: Re: Help: Data Recovery Found this recently.

The text of the poem follows:

```
<> !*"#
^'"$$-
!*=@$_
%*<> ~#4
&[]../
|{,,SYSTEM HALTED
```

The poem can only be appreciated by reading it aloud, to wit:

Waka waka bang splat tick tick hash,
Caret quote back-tick dollar dollar dash,
Bang splat equal at dollar under-score,
Percent splat waka waka tilde number four,
Ampersand bracket bracket dot dot slash,
Vertical-bar curly-bracket comma comma CRASH.

-Tai

From: tep@galt.sdsc.edu (Tom Perrine)
Subject: Re: Oh my god, I hate it...

Perry Rovers <Perry.Rovers@IAE.nl> wrote:

> Here's where you're wrong. Joe Random Hacker was
> pissed because your system was messed up so he
> couldn't use it to leech warez/p0rn, spam and
> modify www.fbi.gov. So he fixed it for you. HTH.
> HAND.

Well, actually, There Was This Site about 2 years ago... Seems that while an outside consultant/security-BOFH was helping them recover from an incident, he discovered that the intruders had gotten pissed off because one of the machines was rebooting on them so often, so they installed about 30 patches[1], a sendmail upgrade[2], backed-up, re-partitioned the disks, and reloaded all the data[3]. They're probably still sad that they actually found the intruders, because they haven't had an OS upgrade since[4].

--tep

[1] including security patches, plus extra stuff[5].
[2] including fixing all their broken aliases, and cleaning the dead accounts from the passwd file[6]
[3] resulting in about a 15% performance increase
[4] still running SunOS4.1.3
[5] SSH :-)
[6] correctly

Call for Papers: AUUG 2002 Theme: "Measure, Monitor, Control"

The AUUG Annual Conference will be held in Melbourne, Australia, on 4, 5 and 6 September 2002 *(subject to change)*.

The Conference will be preceded by three days of tutorials, to be held on 1, 2 and 3 September 2002.

The Programme Committee invites proposals for papers and tutorials relating to:

- Cluster Computing
- Managing Distributed Networks
- Performance Management and Measurement
- Open Source Systems Administration Tools
- System and Application Monitoring
- Security in the Enterprise
- Technical aspects of Computing
- Networking in the Enterprise
- Business Experience and Case Studies
- Open Source projects
- Business cases for Open Source
- Technical aspects of Unix, Linux, and BSD variants
- Open Systems or other operating systems
- Computer Security
- Networking, Internet (including the World Wide Web)

Presentations may be given as tutorials, technical papers, or management studies. Technical papers are designed for those who need in−depth knowledge, whereas management studies present case studies of real−life experiences in the conference's fields of interest.

A written paper, for inclusion in the conference proceedings must accompany all presentations.

Speakers may select one of two presentation formats:

Technical presentation:

- A 30−minute talk, with 10 minutes for questions.

Management presentation:

- A 25−30 minute talk, with 10−15 minutes for questions (i.e. a total 40 minutes).

Panel sessions will also be timetabled in the conference and speakers should indicate their willingness to participate, and may like to suggest panel topics.

Tutorials, which may be of either a technical or management orientation, provide a more thorough presentation, of either a half−day or full−day duration.
Representing the largest Technical Computing event held in Australia, this conference offers an unparalleled opportunity to present your ideas and experiences to an audience with a major influence on the direction of Computing in Australia.

Call for Papers: AUUG 2002 Theme: "Measure, Monitor, Control"

Submission Guidelines:

Those proposing to submit papers should submit an extended abstract (1−3 pages) and a brief biography, and clearly indicate their preferred presentation format.

Those submitting tutorial proposals should submit an outline of the tutorial and a brief biography, and clearly indicate whether the tutorial is of half−day or full−day duration.

Speaker Incentives

Presenters of papers are afforded complimentary conference registration.

Tutorial presenters may select 25% of the profit of their session OR complimentary conference registration. Past experience suggests that a successful tutorial session of either duration can generate a reasonable return to the presenter.

Please note that with the GST changes to tax legislation we will be requiring the presentation of a tax invoice (which we will assist in producing) containing an ABN for your payment. If that is not provided then tax will have to be withheld from your payment.

Important Dates

Abstracts/Proposals Due	− 10 May 2002
Authors notified	− 7 June 2002
Final copy due	− 6 July 2002
Tutorials	− 1−3 September 2002
Conference	− 4−6 September 2002

Proposals should be sent to:

AUUG Inc.
PO Box 366
Kensington NSW 2033
AUSTRALIA

Email: **auug2002prog@auug.org.au**

Phone: 1800 625 655 or +61 2 8824 9511
Fax: +61 2 8824 9522

Please refer to the AUUG website for further information and up−to−date details:

http://www.auug.org.au

Public Notices

Upcoming Conferences

January 28–29, 2002
FAST – First Conference on File and Storage Technologies
Monterey, California

February 6 – 9, 2002
linux.conf.au
University of Queensland,
Brisbane, Australia

February 11–14, 2002
BSDCon 2002
Cathedral Hill Hotel
San Francisco, Californi

June 9–14, 2002
2002 USENIX Annual Technical Conference
FREENIX submissions deadline: November 12, 2001
General Session submissions deadline: November 19, 2001
Monterey Conference Center
Monterey, CA

My Home Network (November 2001)

By: Frank Crawford <frank@crawford.emu.id.au>

Christmas is coming and so are lots of little packages, however, when you look at email, such packages are more likely to be bad than good. What I'm talking about is viruses, worms and trojans, i.e. all those nasties you keep reading about in the PC world. While those with Open Source OS's may not fear these much (which you should, as there are some worms, etc, starting to appear for Linux, etc.), in a home network, it is getting more and more important to do something about them.

Stepping back a bit, as I've previously written, I have a number of machines that run various Microsoft OS's, and use one of my Linux systems as a fileserver (via Samba). As well, the server also acts as a mail server, with a variety of mail readers accessing it via IMAP. Finally, I'm also using Squid as a web cache.

To fight viruses it is necessary to establish "defense-in-depth", i.e. don't just rely on a single protection mechanism, but rather, put in a number of barriers. In my case this means something to scan incoming traffic, particularly mail and web traffic, scanning of the fileserver, and scanning of the desktops. I've had virus scanners on my desktops for as long as I can remember, but up until now, I've done nothing about my other defenses. Well, now the time has come to do something about it.

Unfortunately, there is really no Open Source virus scanning utilities (well, not quite true, but even the open source projects admit they can't keep up - see www.openantivirus.org) so it is necessary to find at least one commercial product. While most antivirus vendors have some form of Linux support, it is often difficult to find and usually aimed at large organisations rather than home users.

After some searching, I selected McAfee's 'uvscan', mainly due to familiarity with their other products. I downloaded a trial beta version, installed it and started playing. Since this is a trial version, it has a 30 day limit (which as of writing I hadn't yet reached). However, I quickly found one problem, how to purchase a licensed version once the time runs out. There seems to be no way to buy it online, and no references to anyone selling it in Australia. I'll let you know how I go in a later column.

Anyway, installation was simple, untar and then run the install script. As with all anti-virus software, it comes with out of date virus definition files, so first step was to download new definition files. This was my main reason for selecting McAfee, I know where to find these definition files, and it is relatively easy to script and automate it. One script I picked up is below, although I have hacked it a bit to make it a better at handling errors:

```
#!/bin/sh
# Script for NAI (McAfee) uvscan by Matt Burke
export PATH=$PATH:/usr/local/bin

wget_args=--passive-ftp

datdir="ftp://ftp.mcafee.com/pub/datfiles/english/
"
uvdir=/usr/local/uvscan
ldatdir=$uvdir/DATs

cd $ldatdir

rm -f .listing*

file=`wget -qnr $wget_args $datdir && grep tar
.listing | awk {'print $4'} | tr
+-cd '[\-.0-9a-zA-Z]'`
if [ -z "$file" ]; then
    echo "$0: No tar DAT file available!"
    exit 1
elif [ ! -f $file ]; then
    wget -q $wget_args -O $file $datdir/$file
    tar --overwrite --directory=$uvdir -xf $file
    echo "$0: DAT file installed: $file"
    uvscan --version
else
    echo "$0: No change to current DAT file:
$file"
fi

exit 0
```

The next step was to test it with a "virus", and then scan the entire system. The need to be able to test anti-virus software is well understood, and there is a pattern defined for such testing, called 'EICAR', which is given in the documentation. This can be cut and pasted to the file 'EICAR.COM' and upon scanning it should be recognised by the scanner.

After this, it is a simple case of setting up a cron job to scan the whole system. Currently I do this weekly, with the following:

```
#! /bin/sh

exec /usr/local/uvscan/uvscan --ignore-links --recursive --exclude
+/usr/local/uvscan/exclude.txt --summary /
```

There are a few problems with 'uvscan', especially, it prints complaints about devices and FIFOs, so they need to be excluded via an exclude file, '/usr/local/uvscan/exclude.txt' in my case. This is just a list of files and directories to exclude.

This, however, is only the first step, and really the most basic. It only allows viruses to be found after they have slipped in. There are at least three additional steps:

- scan mail,
- scan web pages, and
- scanning files on access.

Right at the moment I'm in the middle of implementing 'AMaViS', probably the standard in Open Source mail scanning. It links in with a number of MTA, in particular "sendmail", and with anti-virus software, including "uvscan". If you need more information right now, you can look up http://www.amavis.org but if not, I'll go through the details in my next column. I will warn you, the documentation on the AMaViS site is a bit difficult to get through, primarily because it offers too many

choices and options, i.e. lots of MTAs and lots of anti-virus scanners.

Moving past mail, the next issue is scanning incoming web pages. There are a few projects and one in particular I'm looking at is "Viralator" (http://viralator.loddington.com). I will look into this as soon as I finish with AMaViS. I'll then follow this with online file scanning, and especially a project called "samba-vscan". Unfortunately, this currently doesn't support 'uvscan'.

Once all this is implemented, it will leave two holes, one very traditional, and one rapidly increasing in risk. The first is scanning FTP files, and the second scanning incoming files through various Instant Messaging clients. Scanning FTP is not a big issue, but the IM clients are (this has been where most viruses have been caught lately by my desktop scanners).

With both of these, the issue is a single point of access at which to do the scanning. Unlike mail or web traffic, there is no proxy used for the transfer, rather they are direct connections. If anyone has any suggestions about how to scan these I'd appreciate hearing about it.

Finally, I started out talking about Christmas, and as it is coming up, I'd like to wish you all the best for both Christmas and the New Year, and I hope you get all the presents you wish for.

AUUG Corporate Members

as at 1 July 2001

- Andersen Consulting
- ANSTO
- Aurema Pty Ltd
- Australian Bureau of Statistics
- Australian Industry Group
- Australian Taxation Office
- Australian Water Technologies P/L
- BHP Information Technology
- British Aerospace Australia
- Bureau of Meteorology
- C.I.S.R.A.
- Cape Grim B.A.P.S
- Central Queensland University
- Central Sydney Area Health Service
- Centrelink
- CITEC
- Commercial Dynamics
- Commonwealth Steel Company
- Computer Science, Australian Defence Force Academy
- Computing Services, Dept Premier & Cabinet
- Corinthian Industries (Holdings) Pty Ltd
- Corporate Express Australia Limited
- Crane Distribution Limited
- CSC Australia Pty. Ltd.
- CSIRO Manufacturing Science and Technology
- Curtin University of Technology
- Cyberscience Corporation Pty. Ltd.
- Cybersource Pty. Ltd.
- Daimler Chrysler Australia – Pacific
- Dawn Technologies
- Deakin University
- Department of Defence
- Department of Land & Water Conservation
- Energex
- eSec Limited
- Everything Linux & Linux Help
- Fulcrum Consulting Group
- Fulcrum Consulting Group
- G.James Australia Pty. Ltd.
- IP Australia
- IT Services Centre, ADFA
- Land and Property Information, NSW
- LPINSW
- Macquarie University
- Mercantile Mutual Holdings
- Motorola Australia Software Centre
- Multibase WebAustralis Pty Limited
- Museum Victoria
- Namadgi Systems Pty Ltd
- Nokia Australia
- NSW National Parks & Wildlife Service
- NSW Public Works & Services, Information Services
- Peter Harding & Associates Pty. Ltd.
- Qantas Information Technology
- Rinbina Pty. Ltd.
- Security Mailing Services Pty Ltd
- Snowy Mountains Authority
- St. John of God Health Care Inc.
- St. Vincent's Private Hospital
- Stallion Technologies Pty. Ltd.
- Standards Australia
- State Library of Victoria
- TAB Queensland Limited
- The University of Western Australia
- Thiess Contractors Pty Ltd
- Tower Technology Pty. Ltd.
- University of Melbourne
- University of New South Wales
- University of Sydney
- University of Technology, Sydney
- Victoria University of Technology
- Westrail
- Workcover Queensland

Review: SuSE Linux Professional 7.3

Author: Layne Heiny <lph@linuxtests.org>

Our copy of SuSE Linux Professional 7.3 arrived Friday October 26, 2001. We ordered it directly from SuSE by using their webstore.

For this review we will list several of our installation and post-installation experiences. The review cannot be complete because we've only worked with the distribution for two weeks. We've also spent very little time with different hardware configurations (2 different machines, three video cards, memory configurations, etc). Plus, SuSE includes thousands of applications and it would be impossible for the three of us to try all of these before writing about our experiences. Therefore, we've chosen five major tasks (or jobs) to include in this review.

Our first job was obvious after reading the advertising on the box. The statement that caught our attention was, "A ready-to-use system in 20 minutes!" (*depending on hardware). Could that really be possible? Our installation experiences with previous releases were never that fast.

Our second job also became clear after looking at the packaging and reading some of the marketing literature; is SuSE 7.3 "child proof" as stated? Those are strong words. Can any Linux OS be "child proof"? What does that mean, anyway?

Third, is SuSE Linux 7.3 "more than a mere server and desktop operating system?" The latest debate among pundits has been whether Linux is a viable desktop OS. Our experiences with KDE and GNOME support the idea that Linux is a desktop OS - and there isn't a debate. We also tried blackbox with this version of Linux.

Fourth, should you upgrade from SuSE 7.1 or SuSE 7.2?

Finally, we were interested in how the package operated from the perspective of a new user. Would we get lost from the moment the OS was installed? Would there be plenty of help? Would we be able to get problems resolved easily and with little frustration? Does documentation match what we see on the screen? And many other things that old timers forget.

Specifications

- Kernel: 2.4.10
- glibc: 2.2.4
- XFree86: 4.1.0
- KDE: 2.2.1
- GNOME: 1.4.1 Beta 2 including Nautilus
- Filesystems: FAT, ext2, ext3, ReiserFS, NTFS, and JFS (JFS and ext3 are new)
- Networking: TCP/IP, IPsec, NFS, ADSL, IP v6, PPP, IPX, netatalk, ISDN, Samba, Token Ring, ARCnet, Ham Radio, UUCP, SLIP
- Networking Software: Apache, Squid
- Browsers: Konqueror, Lynx, Mozilla, Netscape 6.1, Opera
- Mail: Sendmail, Postfix
- Databases: MySQL, PostgreSQL
- e-Business: DB4web
- Office Applications: Acrobat Reader, KOffice 1.1, StarOffice 5.2
- HTML editors: Bluefish, Quanta +
- Image Processing: GIMP 1.2.2, ImageMagick, Moonlight 3D
- Audio/Video editing: Broadcast 2000, Mainactor.
- Sound: CD Player, MIDI tools, MP3 player, MOD player, RealPlayer, Xmms

New Features of SuSE 7.3

New modules in YaST2

- Scanner configuration
- Configuration tool for the SuSE Firewall (?)
- LVM configuration: partitioning while the system is active
- Setup of TV cards (bttv)
- Automatic detection and configuration of the majority of IDE-CD writers
- Runlevel editor for activating/deactivating system services
- Keyboard selection while the system is active
- Basic X window configuration in YaST2 while X is active (with ISaX)
- NIS server administration tool

YaST2 improvements

- The partitioning tool can be accessed with YaST2 while the system is active
- The partitioning tool makes a proposal for the optimum organization of the partitions.
- New package selection: packages are grouped and groups can be combined.
- Support of the new file systems ext3 and jfs
- New module for software RAID support
- ncurses shortcut support in YaST2 text mode
- Graphical YaST2 now available within VMware systems

SaX2 improvements

Configuration of touchscreens and graphics tablets
"Cloned multihead" support (combination of several screens on a single screen).
Quick configuration of devices, desktop settings, and multihead while the X server is active

New features in the manuals

Description of the 3D-acceleration of your graphics card
New documentation on SaX2 and its new features
Completely revised TV chapter

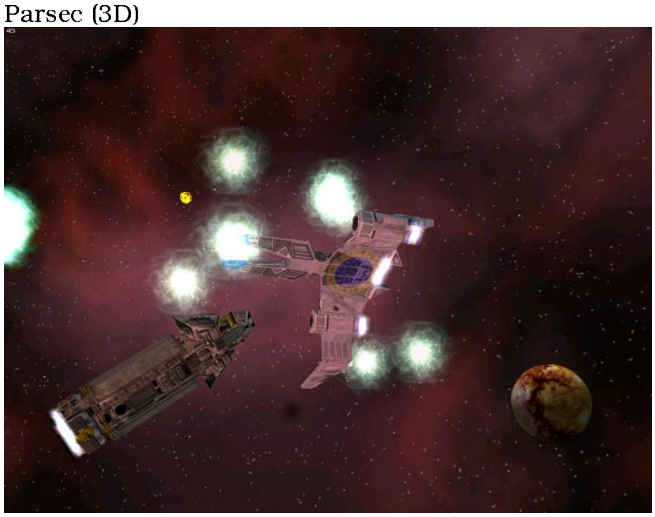
Additional new features and improvements

- New comfortable scanner program kooka
- NIS and LDAP client

- Optimized linker: through intelligent arrangement of program functions in the so-called relocation tables, an efficient cache can be made use of, thus speeding up the start-up of many programs (especially large programs).
- New TV applications

Many fascinating new games...

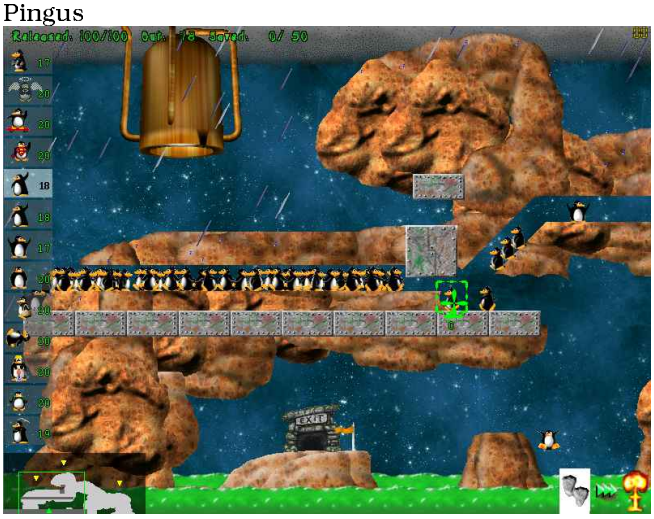
Parsec (3D)



Tuxracer (3D)

FlightGear (3D)

Pingus



And a fair few more, including:

- Descent
- Armagetron (3D)
- Cannonsmash (3D)
- Chromium B.S.U. (3D)
- Clanbomber

More new features

- YaST2 configuration for BTTV cards and scanners
- KDE 2.2.1 with Kooka, Kate, SuSE desktop, Kdict
- KOffice 1.1
- SaX2 support for Touch Screens
- Kernel 2.4, glibc 2.2.4, XFree86 4.1.0, SoftRAID, ext3, and JFS
- YaST2 for LVM (logical volume manager)

We did not play with the scanner configuration (despite having a scanner hooked up), kooka nor TV cards.

Unique Features of SuSE 7.3

SaX2, YaST and YaST2
It should be stated that some people object to SuSE due to the licensing issues of YaST and YaST2.

Advertising on SuSE website:

We thought it would be appropriate to list the way SuSE is advertising this product.

- SuSE Linux 7.3 is an operating system for the whole family. linux for the kids? No problem: Child's play: with the graphical user interface KDE, using SuSE Linux 7.3 is mere child's play.

- Child-proof: no system file can be damaged accidentally or purposely.

- Playful: SuSE Linux 7.3 comes with a large number of games, the games series has been considerably expanded and improved. Multimedia applications like TV in Linux and sound applications, too, have been substantially expanded and improved.

- Multimedia: writing CDs, watching and processing videos, mixing sound, an editing studio, and a synthesizer are only some of the many multimedia options.

- Privacy: each user has his own private environment that can't be viewed, modified, or deleted without his consent.

- Network: set up your own family network - a simple task for SuSE Linux, the par excellence network operating system.

Installation

After one week, we managed to install SuSE Linux Professional 7.3 about twenty times, maybe a few more than twenty because we lost count. The longest installation took place when using the upgrading feature.

Our second week with the product was spent installing the distribution on a different machine and then trying to actually use the product.

Our first installations were done on our test system used in other reviews. We temporarily hooked up our model WD300 ATA-100 hard drive on to the Tyan motherboard. During some of the tests, we temporarily changed the video card to our Voodoo 3000 card after running into trouble with our nVidia based video card.

LinuxPlanet is publishing problems with the "update" feature of 7.3. We have done the update from SuSE 7.1 to 7.3 and experienced a few problems but nothing in comparison to LinuxPlanet. This may be due to our configurations.

The final testing took place on an AOpen AX3S motherboard, which includes the i815 video and audio onboard. We thank ESC Technologies for their support and sending us this configuration..

Descriptions of Installations

For our first installations (the Tyan system), we used the floppy and DVD supplied in the package; that is, we booted off the floppy diskette, started the install and within 14 minutes had a working copy of Linux.

There were many changes to the installation screens. However, since we didn't play with 7.2 - we aren't sure which ones were implemented in this version. So, please excuse us for our ignorance. You'll want to read other reviewers works to find the answer to the screen changes.

During the installation on the AOpen system, we booted from the DVD and saw a few different screens. Because of the difference in video, we also did not see the slide show on the AOpen system.

After purchasing a new MAG monitor tonight (model 96FS), we decided to reinstall SuSE 7.3. We ran into a problem with screen resolutions. During the install, we were only given the choice of 800x600. After running YaST2, checking the hardware information, the information for monitor and video chipset (i815) were correct. We re-ran X11 configuration and one screen showed 1280x1024 but the combo box only allowed 800x600 and 640x480 as choices. Upon reboot, the machine entered into 1024 resolution. Very strange indeed.

Below is a set of different types of installs that we performed.

- A second drive with Windows MEOur first installation was a two hard drive setup. The second drive had Windows ME and we installed SuSE Linux 7.3 on the first drive. We simply followed the defaults.

- A fresh install choosing options during the install.

- If you prefer to manually change settings during the YaST2 installation then there are plenty of options. For example, during several of the installs, we chose to install Ext3. This was simply a matter of entering the options during the install. It was very straight forward.

- An upgrade of SuSE 7.1 This was done as a fresh install of SuSE 7.1 on our hard drive and then did a SuSE 7.3 upgrade.

EXT3

A new feature for SuSE 7.3 is Ext3. Our final installation used Ext3. Soon after the release of this distribution SuSE released a patch: (http://sdb.suse.de/en/sdb/html/ext3_rootfs_73.html).

as a 1.0 version. A release candidate is available on the Ximian website. A beta is available on the DVD.

We were able to download messages from our server; however, we were not able to create new messages or reply to messages.

Installation of KBear
We usually use gFTP but decided to explore KBear 1.2.1. This application is on the DVD and was installed using YaST2.

Installation of GNOME
GNOME is not installed by default. However, many users of Linux prefer this environment.

Installation of Blackbox
Since the WindowManager Blackbox was not installed during the default, it was installed through YaST2. Immediately after the installation, entering Blackbox from the login screen was available. However, after shutting down and restarting Linux, one of the users could only get into Blackbox. Even choosing KDE or GNOME took the user into Blackbox. In order to fix this problem, we deleted the user and created a new one for that user.

Installation of fvwm2
We decided to try this WindowManager on our AOpen system, in hopes that the system ran more smoothly (KDE2 and GNOME were choppy at best). Indeed, this is a perfect WindowManager for a 64 MB system.

Installation of gFTP
This is our preferred FTP applicaiton. Installation was through YaST2.

Installation of Mozilla, Opera, Galeon, and Netscape 6
The most exciting part about Linux is seeing all of the advances being made in so many software applications. The browser market is set for the Windows OS, however, there are many choices under Linux.

Installation of Xine
Xine is included on the DVD. If you have a video card that can handle 3D acceleration then this is a great way to watch DVD movies on your PC. We initially only saw the slide show effect for our 3D applications. However, after installing the driver from nVidia, we were able to watch movies. We also needed to set the dma for the DVD drive. Directions are on the Xine website.

Installation of Ximian Desktop
We tried to install Ximian Desktop. This option is not available for SuSE 7.3 – but 7.2 is availble and therefore we chose that option (silly, right?). Upon running lynx -source http://go-gnome.com/ | sh the download started. However, a few minutes into the download, we crashed with the statement that there wasn't a display. We exited and tried to enter GNOME 1.4. There wasn't a display ;-) At this point we were stuck and couldn't get to a prompt. We thought this would be an excellent opportunity to try

SuSE WEBSITES

A lot of effort has gone into the SuSE websites. We constantly use the English version.We searched the Knowledge Database for help on our blackbox problem. However, the only problem in the database was regarding SuSE 7.2. Unfortunately the solution didn't help us.

FAQs AND KNOWLEDGE BASE

This is probably one of the most important places on the SuSE website. However, EXT3 is not explained on the website. It would be a nice whitepaper or a pointer to where to get informa

SUGGESTIONS FOR IMPROVEMENT

Information on ext3 and ReiserFS is missing in the documentation. In fact, none of the manuals mention ext3. Since these are fairly new features, users would look to the documents for details.

YaST2 would be greatly improved if the version number of the application was included in the same screen as the size and short description. Presently, the user must click on the description button to get this information.

CONCLUSIONS

In our opinion, SuSE Linux Professional 7.3 is an excellent product. The installation is fast. Hardware was detected correctly.

For the first time in our website's history, all three of us have come down on the side of SuSE. Generally one of us is in favor of SuSE – no matter what they publish. Another one is a Mandrake user while the third one can't seem to get any Linux distribution to work as he'd like.

Relative to our five jobs,

Does SuSE install quickly?
Yes!

Is SuSE child proof?
No. In our opinion, this was stupid. There is no such thing. Children have an incredible way of thinking differently from adults. Any adult trying to outsmart a child is just silly.

Is SuSE more than a mere desktop?
Well, not exactly. It's still a step forward in the right direction but let's not go overboard. Our main hesitation is the amount of hardware necessary. Our 64 MB system ran too sluggish under KDE 2. Instead, change to a different WindowManager – which may not be easy for new users. We'd recommend using blackbox or fvwm2.

Should you upgrade to SuSE 7.3 from a previous version?

Our experiences suggest that users should upgrade.

Is SuSE appropriate for new users?
Now this is where SuSE shines! Yes, users purchasing a new computer and new to Linux should absolutely try SuSE 7.3 Either have your OEM install it for you or install it yourself.

Old timers who are trying to stretch the life out of an old machine should hesitate. Upgrade the memory first.

Corrections and Additions

After publishing this article, we will list changes suggested by our readers in this section. You may send feedback to us at webmaster At this time, no one has said "boo."

Other Resources

- SuSE website: http://www.suse.com/
- FirstLinux (http://www.firstlinux.com/) and
- LinuxPlanet (http://www.linuxplanet.com/) have also written reviews and experiences.
- Linux Journal (http://www.linuxjournal.com/) has awarded SuSE Linux 7.3 Product of the Year

This article is re-printed with permission. The originals can be found at:

http://www.killertux.com/modules.php?op=modload&name=Reviews&file=index&req=showcontent&id=4

PDF Service with Samba

Author: John Bright < jbright@winfordeng.com>

Introduction

PDF documents provide a great way to pass around documents on the Internet. They have many uses, such as sending quotes and invoices to business clients. Two of the main reasons the PDF format is so popular is that it preserves all of the document's formatting exactly and it is easily viewable on almost all platforms. For many computer users stuck in the Windows paradigm, creating PDF documents means forking over precious cash to the folks at Adobe. However, this article will show you how to use Linux, Samba, and Ghostscript to provide a PDF creation service to both Windows and Linux users. Of course, all of this can be obtained for free.

First, let's take a look at the overall scheme of operation. We will use Samba to provide a "pseudo-printer" service (it will look like a standard printer to clients) that will use Ghostscript to create a PDF document out of any Postscript printer job that is queued onto it. We will then configure the Windows machines to use this shared printer and send jobs to it in Postscript form.

Samba

Samba is a great piece of software that runs on Linux/UNIX and allows you to share files and printers with Windows machines. Samba provides services that are compatible with the standard "Windows Networking" services provided by Windows 95/98/NT/etc computers. Before we get into configuring Samba for our purposes, you'll need to make sure that the Samba server is installed on your Linux system. As always, you can download the Samba source from www.samba.org, but generally the easiest way to install it on your system is by installing the "samba" package provided by Debian, Red Hat, or whoever.

If this is your first time installing Samba, you will want to review/edit some of the basic configuration options in the smb.conf (look in /etc or /etc/samba) configuration file. The main things to watch in order to get your services up and running are the security policy (security=share or security=user) and the "guest account" setting. For details of configuring Samba, refer to the Samba documentation at www.samba.org or the SMB HOWTO. A complete sample (low-security) configuration file will be shown later.

It is probably advisable to test your connection and authentication method (if any) by creating a simple file share for your clients. In any event, once your clients are able to connect to your Samba server, we are ready to create the PDF "pseudo-printer". First,

though, let's make sure we have the right utilities to actually produce the PDF documents.

Ghostscript

Ghostscript is another great application that can be used on a Linux system. Ghostscript is often used to convert Postscript into the correct raw format for a printer, but it can also be used to convert between Postscript and PDF formats. Ghostscript comes installed on many distributions in order to provide printer support. If the "gs" command is available on your system, then Ghostscript is probably already installed. Otherwise, you can install your distribution's package (ghostscript on Red Hat, gs or gs-aladdin on Debian) or download the source from http://www.cs.wisc.edu/~ghost/ if you're feeling adventurous.

The Ghostscript package includes a script called ps2pdf that makes the conversion of Postscript to PDF quite easy. Now that we have this utility available, we can begin the creation of our PDF service on Samba.

Bringing It Together

First, let's review a typical bare-bones printer share in Samba (from the smb.conf file):

```
[hpdeskjet]
path = /tmp
printable = yes
writeable = no
create mask = 0700
guest ok = yes
printer name = lp
```

(Note the silent "e" in writeable. The configuration file has it even though the ordinary word doesn't. The same applies to browseable below. Actually, Samba accepts it either way, but Samba's manpages use writeable.) Normally, when a print job is spooled to this share, a command such as lpr is run to transfer the job to the Linux printing system. Our method here is to use the excellent configurability of Samba to specify an alternate printing command in place of lpr. Specifically, the configuration variable is called "print command".

The specified command is executed, and any occurrence of %f or %s in the "print command" variable will be replaced by the name of the printer spool file that was sent in by the windows client. For example, to simply discard any print jobs, this line could be placed in the above printer configuration:

```
print command = /bin/rm %f
```

This brings up another important point: whatever print command is specified must delete the spool files, or else they will eventually pile up and fill your disk.

Print Script

Our print script will accept one command-line argument: the name of the print spool file, which is assumed to be in Postscript format. It will then convert this into the PDF document and place it in an accessible location. Clients will be able to retrieve the finished product by using the file sharing services of Samba. For example, if a directory named "/shr" is shared by Samba, we could place finished PDF documents in `/shr/pdfdropbox/`. Be sure to `mkdir` whatever directory you choose.

Also, you must be sure that you give write permission to the Samba user (the nobody user in this example) or it will not be able to create any PDFs. In this example, you would want to:

```
chown nobody /shr/pdfdropbox
chmod u+rwx /shr/pdfdropbox
```

Here is the complete, yet simple print script, called printpdf, also available in text format. On our Linux system, we'll place the script at `/usr/bin/printpdf`

```
#!/bin/sh # Simple script to convert a specified postscript file
into a PDF document
# and place it in a location that is shared by the Samba server.
#
# Arguments:
#   1st - The name of the spool file
#
# John Bright, 2001, jbright@winfordeng.com # We will create the
pdf into a temporary file based upon the current date and time.
# After we are finished, we'll rename it to a file with the same
date, but ending
# in .pdf.  We do this because if a user tries to open a PDF that
is still being written,
# they will get a message that it is corrupt, when it is actually
just not done yet. DATE=`date +%b%d-%H%M%S` # Directory in which to
place the output
# Be sure this directory exists and is writable by the user that
Samba
# is running as (for example, the nobody user)
OUTDIR=/shr/pdfdropbox ps2pdf $1 $OUTDIR/$DATE.temp
mv $OUTDIR/$DATE.temp $OUTDIR/$DATE.pdf
rm $1
```

I said it was simple, right? There's really not much to it once we have all of the tools together.

Finish the Samba Setup

Now that we have seen everything that goes into the PDF service on the Linux side, we can finish the Samba configuration file. Here is an example smb.conf file that gets the job done. It is a little low on security, but that keeps everything simple. You can download this file from here.

```
[global]
   guest account = nobody
   invalid users = root ; Tighten security just a little: only
allow local access
   interfaces = 127.0.0.1 eth0
   bind interfaces only = Yes
   ; This assumes you are on a local network with 192.168.x.x IP
addresses
   hosts allow = 192.168.
; Share-level security is generally easier, although not as secure
   security=share
   workgroup=WORKGROUP ; Set up a public share, this will be used
to retrieve PDFs
; The name of the share will be seen as "shr" by Windows users
[shr]
   path = /shr
   browseable = yes
   writeable = yes
   guest ok = yes
   force user = nobody ; Set up our PDF-creation print service
[pdf]
   path = /tmp
   printable = yes
   guest ok = yes
   print command = /usr/bin/printpdf %s

   ; There is no need to support listing or removing print jobs,
   ; since the server begins to process them as soon as they
arrive.
```

```
   ; So, we set the lpq (list queued jobs) and lprm (remove jobs in
queue)
   ; commands to be empty.
   lpq command =
   lprm command =
```

Of course, you will need to start/restart Samba after you have created/edited the smb.conf configuration file to your liking.

Setting Up a Windows Client
You should now be able to go ahead and install the shared PDF printer as a network printer on your Windows client machine. To do this, find the printer share under Network Neighborhood, right-click, and select Install. During installation, you will be asked to pick a printer driver. Just select some Postscript printer driver, for example, the HP LaserJet 5P/5MP PostScript.

To briefly explain how this is fitting together, the PDF service on your Linux machine is expecting to receive input in Postscript format. Since our printpdf script receives the print job exactly as it was sent by the Windows client, this means we need to have the Windows clients send print jobs in Postscript form. As described above, this is done by selecting a driver for any Postscript printer on the Windows client when the PDF network printer is installed. I generally select some variety of HP Laserjet PS printer from Windows' printer driver list (such as the HP LaserJet 5P/5MP PostScript, as noted above) although it doesn't matter a whole lot because all of the Microsoft-supplied Postscript drivers use the same core driver to generate the Postscript.

Once you have the PDF network printer installed on your Windows machine, simply print anything from any program to your new network printer, and you should have a PDF document waiting for you shortly.

Streamlining

If you have an office full of non-computer-savvy folks, it would probably be more trouble than it's worth to try to have them go through the installation process and select an appropriate printer driver. If you have ever installed a network printer from another Windows machine, you have probably noticed how the printer driver is automagically copied to your machine so that you are never even prompted for a driver. We can do the same thing with Samba. First, you should set up a file share on your Linux machine named "printer$" (without the quotes). We'll make the path for the printer$ share be `/etc/samba/printdrivers/` (you'll have to mkdir the directory). The clients will simply use this share to obtain the printer driver files during installation.

Now we need to find out which driver files must be copied into the printer$ share directory. We also need to give Samba a printer definition so it can tell the client which driver files it needs. It turns out all this is taken care of in one step thanks to a Samba utility called make_printerdef. This utility requires you to have the Windows INF file that defines your printer and know the full title, such as "HP LaserJet 5P/5MP PostScript". You will need to find which INF file your printer is defined in. For example, this LaserJet is defined in `C:\WINDOWS\INF\MSPRINT3.INF`. Note that `C:\WINDOWS\INF` is a hidden directory.

Copy this file onto your Linux machine and use the make_printerdef utility to create a local printer definition file that Samba will read. For example:

```
make_printerdef MSPRINT3.INF "HP LaserJet 5P/5MP
PostScript" >> /etc/samba/printers.def
```

Here we redirected standard output to the printers.def file to create the printer configuration. The make_printerdef program also outputs some explanation on standard error which you will see. It should tell you which driver files you need. You can find these in `C:\WINDOWS\SYSTEM` or `C:\WINDOWS` and you should copy them into the path of your printer$ share on the Linux machine (in our case, `/etc/samba/printerdrivers/`).

The printers.def file that we have created (or appended to) here does not need to be shared to the Windows machines, it is only read by Samba. Now we just have to tell Samba about the printers.def file and our driver files. This is done with the "printer driver file" setting in the global section and the "printer driver" and "printer driver location" settings in each printer section of smb.conf. The following revised smb.conf file shows how these settings are used, and also shows an example of a printer$ share. You can download this configuration file from
here.

```
[global]
   guest account = nobody
   invalid users = root    ; Tighten security just a little: only
allow local access
   interfaces = 127.0.0.1 eth0
   bind interfaces only = Yes
   ; This assumes you are on a local network with 192.168.x.x IP
addresses
   hosts allow = 192.168.

   ; Share-level security is generally easier, although not as
secure
   security=share

   workgroup=WORKGROUP    printer driver file =
/etc/samba/printers.def

; Set up a public share, this will be used to retrieve PDFs
; The name of the share will be seen as "shr" by Windows users
[shr]
   path = /shr
   browseable = yes
   writeable = yes
   guest ok = yes
   force user = nobody ; Set up our PDF-creation print service
[pdf]
   path = /tmp
   printable = yes
   guest ok = yes
   print command = /usr/bin/printpdf %s

   ; There is no need to support listing or removing print jobs,
   ; since the server begins to process them as soon as they
arrive.
   ; So, we set the lpq (list queued jobs) and lprm (remove jobs in
queue)
   ; commands to be empty.
   lpq command =
   lprm command =

   ; We already defined the printer driver definition file above.
   ; Here we need to specify the entry in that file that should be
used
   ; for this printer.
   printer driver = HP LaserJet 5P/5MP PostScript
   printer driver location = \\%h\printer$ ; File share to allow
clients to download printer drivers
[printer$]
   path = /etc/samba/printdrivers
   guest ok = yes
   read only = yes
```

Setting Up a Linux Client

This section isn't necessary for providing a PDF service to Windows clients. This section describes the procedure for using the PDF service from Linux clients.

The service probably isn't quite as useful for the Linux clients, since they can more easily install all the necessary tools on their own machines, but it still might be useful to have a centralized PDF creation service. (Side note: Ghostscript is available for the Windows platform, but most users would probably find it quite difficult compared to the printer service-based technique described here.) Also, the technique used to print to the PDF service can be used to print to any other printer service shared by Samba or Windows, so it is good information to cover.

There are numerous ways that you can print to a Windows printer share from Linux. Probably the best is to list the smbprint script (which uses smbclient) as a filter in an `/etc/printcap` entry. When this method is used, a Windows shared printer can be used with the standard lpr command that Linux users and applications are accustomed to. You will need to make sure you have both the smbprint and smbclient programs on your computer. The smbclient program is in the "smbclient" package on Debian systems and the "samba-client" package on Red Hat systems. On Red Hat, the smbprint script is included with the "samba-client" package. On Debian, it is included with the "samba-doc" package as well as a different version in the "printfilters-ppd" package and "lprngtool". There are so many different versions floating around that I thought it best to include a copy here. You can download it from here. In any event, I'll assume that you have a working smbprint at `/usr/bin/smbprint` and that it is executable (`chmod +x /usr/bin/smbprint`). Here is the smbprint script:

```
#!/bin/sh
# This script is an input filter for printcap printing on a UNIX
machine. It
# uses the smbclient program to print the file to the specified
smb-based
# server and service.
# For example you could have a printcap entry like this
#
#
smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# which would create a UNIX printer called "smb" that will print
via this
# script. You will need to create the spool directory
/usr/spool/smb with
# appropriate permissions and ownerships for your system. # Set
these to the server and service you wish to print to
# In this example I have a Windows for Workgroups PC called
"lapland" that has
# a printer exported called "printer" with no password. #
# Script further altered by hamiltom@ecnz.co.nz (Michael Hamilton)
# so that the server, service, and password can be read from
# a /usr/var/spool/lpd/PRINTNAME/.config file.
#
# In order for this to work the /etc/printcap entry must include an
# accounting file (af=...):
#
#   cdcolour:\
#:cm=CD IBM Colorjet on 6th:\
#:sd=/var/spool/lpd/cdcolour:\
#:af=/var/spool/lpd/cdcolour/acct:\
#:if=/usr/local/etc/smbprint:\
#:mx=0:\
#:lp=/dev/null:
#
# The /usr/var/spool/lpd/PRINTNAME/.config file should contain:
#   share=PC_SERVER
#   user="user"
#   password="password"
#
# Please, do not modify the order in the file.
# Example:
#   share=\\server\deskjet
#   user="fred"
#   password="" #
# The last parameter to the filter is the accounting file name.
#   Extract the directory name from the file name.
#   Concatenate this with /.config to get the config file.
#
eval acct_file=\$$#
spool_dir=`dirname $acct_file`
config_file=$spool_dir/.config # Should read the following
variables set in the config file:
#   share
#   hostip
#   user
#   password eval `cat $config_file` share=`echo $share | sed
"s/[\]/\//g"` if [ "$user" != "" ]; then
   usercmd="-U"
else
   usercmd=""
fi if [ "$workgroup" != "" ]; then
   workgroupcmd="-W"
else
   workgroupcmd=""
fi if [ "$translate" = "yes" ]; then
 command="translate ; print -"
else
 command="print -"
fi
#echo $share $password $translate $x_command > /tmp/smbprint.log
cat | /usr/bin/smbclient "$share" "$password" -E ${hostip:+-I} \
     $hostip -N -P $usercmd "$user" $workgroupcmd "$workgroup" \
     -c "$command" 2>/dev/null
```

The next step is to add a new `/etc/printcap` entry and list the smbprint script as a filter.

Here is an example printcap entry (or complete file), also available as a
text file:

```
# PDF Service entry lp|pdf|PDF Printer:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/pdf:\
        :af=/var/spool/lpd/pdf/acct:\
        :mx#0:sh:\
        :if=/usr/bin/smbprint: You will need to create the spool
directory /var/spool/lpd/pdf/
```

(or if you have LPRng, run checkpc -f). Be sure to keep the accounting file line in the printcap entry, and be sure the accounting file is located in the same directory as your .config file, as this is how the smbprint script finds the .config file.

Also, it is standard procedure to have the system's default printer named "lp" as shown above. If you already have a `/etc/printcap` file and would like to retain your existing default printer, you should remove the leading "lp|" from the entry shown above. Next, you need to create a configuration file named ".config". You should create this at `/var/spool/lpd/pdf/.config` The .config file defines which server the print job should be sent to. Here is an example:

```
share=//yourserver/pdf
user=""
password=""
```

Here, yourserver should be replaced by the name of the computer providing the PDF service.

If you have any trouble with this, make sure that the smbprint script has permission to read the .config file, or you may be scratching your head for a while. Probably the safest way, at least at first, is to give read permission to all, for example: `chmod a+r /var/spool/lpd/pdf/.config`

Finally, to print to the PDF service from Linux, invoke the command:

```
lpr -Ppdf file_to_print.ps
```

on a Postscript file. This can also be used from within most applications. For example, listing "lpr -Ppdf" as the print command in Netscape will allow you to

create a PDF document from a web page.

Viewing PDF Documents

The final topic to be covered deals with how to view PDF documents.

Everybody knows the standard on Windows is Adobe Acrobat Reader, but there are many more options on Linux. Unfortunately, none of the current options on Linux seem to be quite as dependable as Reader on Windows, but they are still very workable. The main options are:

- acroread – Adobe has a nice version of Acrobat Reader for Linux
- gv – Viewer that uses ghostscript to interpret the PDF
- gnome-gv – Also uses ghostscript, but has a nicer user interface
- xpdf – A nice lean PDF viewer, but not a fancy interface

In my opinion, gnome-gv has the nicest user interface. It is based on GTK+, so things like the mouse scroll wheel work without any special consideration. Unfortunately, it will fail to read some PDF documents and display a nasty-looking error from ghostscript. From my experience, acroread is very good about being able to interpret documents. In the past

I have had some trouble with it crashing, but I think it has gotten better since then.

I have rarely used gv, but I imagine it has the same problem as gnome-gv since they are both based on ghostscript. Finally, xpdf is a very stable PDF reader. I don't recall every having it crash, and it usually has no problem interpreting documents.

Still, there is an occasional problem, and the displayed quality of the document often isn't quite up to par. It doesn't have a full feature list, but it is a good viewer to keep around. All this may sound scary, but be assured that on average, PDF viewing on Linux is not a problem.

Have fun and good luck!

John Bright is a partner in Winford Engineering and flawlessly performs his assigned programming and Linux administration duties :). He also administers several Linux/UNIX computers at a local university and always has several Linux-related projects to keep him busy.

This article is re-printed with permission. The originals can be found at:

http://www.linuxgazette.com/issue72/bright.html

The Linux Terminal – a Beginners' Bash

Author: Ramon Casha <ramon.casha@linux.org.mt>

In Linux, the command-line terminal offers a wealth of very powerful tools, and is not all that difficult to use. This tutorial explains to ordinary, non-technical users, some of the basics of how to use the shell.

Abstract

This tutorial presents the Linux terminal and the "bash" shell to people who have never used a command line to give commands to an operating system before, or who have never done so in Linux/Unix. People who have already used a Unix shell before might find it a bit simple.

Due to the popularity of the Microsoft Windows operating system, and the large number of ex-Windows-users who have discovered Linux, I have provided comparisons to equivalent or similar features and terminology in Windows' MS-DOS prompt or Command prompt. These are provided as an additional help for Windows users, and are not necessary to follow this document.

Since this tutorial is intended as an introduction, it is purposely not comprehensive. Several commands, for example, are only explained in the depth necessary to gain an understanding of what they do and how to use them, not necessarily to use them to their full potential.

In the article I assume that the user is already familiar with concepts such as files and directories, as well as filenames, etc.

Introduction

Nowadays, as soon as you get Linux installed, you get a nice graphical interface and rarely if ever need to make use of the so-called terminal mode (aka shell prompt).

However, in Linux the simple, modest terminal is not merely an afterthought, but an extremely powerful tool. While it may be true that you don't need to use it, it's not that difficult to learn, and very useful to know. Fortunately for yourself, with Linux's users and security, you can create a new user for playing around, then you can experiment to your heart's content without breaking anything. In this document we will go step by step through many common tasks.

Hopefully by the end you will feel quite familiar with the terminal.

This will come in handy when, for instance, you encounter some document which instructs you to open a terminal and enter certain commands.

Preparation

As I mentioned before, it's a good idea to create a new user for experimenting. In Linux, a normal user cannot break anything in the system, but can still delete his/her own files. By creating a new user you will have a new "playpen" to play around in, so if you make some mistake you won't delete the files you normally work with.

One thing you must never do (until you know what you're doing) is to work as the "root" user. This user has the necessary permissions to delete or alter all files on the computer, which is generally not considered a good move.

So the first thing to do is to use your favourite tool to create a new user. Use all the normal settings, but ensure that the "shell" setting is "/bin/bash". I called mine "sandbox" but you can use any name you like. Since different Linux distributions provide different tools which can be used to perform this step you should use whichever tool you find best. Some examples include KDE's User Manager or the LinuxConf tool.

This picture shows KDE's User Manager being used to create the user "sandbox". Any other user-management program may be used instead.

After you have created the new user, you can log out, then log in using that user to ensure that you can perform any command safely.

The "Bash" shell

A "shell" is a program which interprets commands, either typed in directly by the user, or contained in a file called a "shell script", which is a simple interpreted program. The equivalents in WindowsTM would be "command processor" for shell, "COMMAND.COM" or "CMD.EXE" instead of bash, and ".BAT files" instead of shell scripts. Linux has a variety of different shells, but certainly the most popular is "bash", so it is this one which will be described here (even though many of these instructions apply to all shells). Some of the others are retained simply because there are lots of people who got used to them and don't wish to change, or because they are aimed at a specialised set of users.
Trivia: Linux, like all unixes, uses a lot of acronyms for its program names, many of them somewhat humorous (eg, yacc="yet another compiler compiler"). Most shells end in "sh", and include ksh (korn shell), csh (C shell... seashell, get it?) and bash (Bourne again shell).

Accessing the terminal

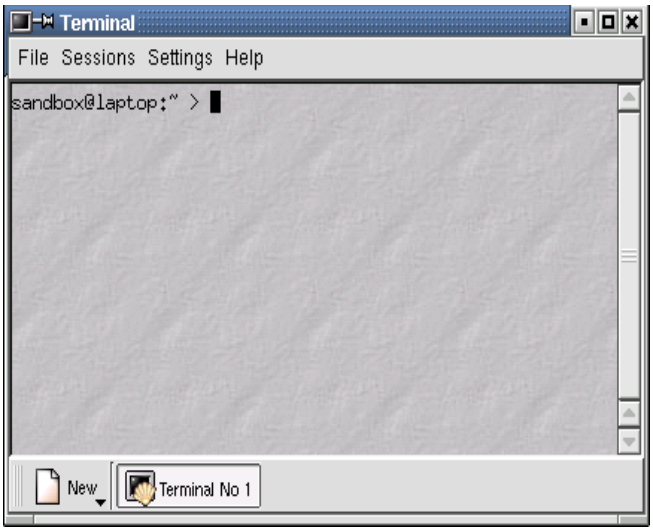
The easiest way to access the terminal, once you have logged in as the new user you created, is to click on the terminal icon from the panel at the bottom of the screen.

The left picture shows KDE's terminal icon, and the one on the right shows Gnome's. The instructions in this document apply to both, since both will load the user's default shell which is bash. With other GUI's, look for "xterm" in the menus, or any other program name ending in "term" or "terminal".

Another way of accessing the terminal is to load Linux in text mode.

Many servers are configured not to load graphics by default, so the first thing you'd meet is a login prompt in text mode. "Home" or "desktop" computers on the other hand usually start up in graphics mode.

You can try the text-mode prompt easily by pressing Alt-Ctrl-F1 to switch to text mode and Alt-Ctrl-F7 to switch back. Actually, from Alt-Ctrl-F1 to Alt-Ctrl-F6 are usually set up as 6 individual text-mode login prompts, while Alt-Ctrl-F7 is the graphics mode.



This picture shows a terminal window – in this case, KDE's "Konsole".

When you first load the terminal window (or log in from the text-mode login prompt), you will see a prompt somewhat like the above. The actual text within the prompt can be user-defined, and varies between distributions. Generally however, the prompt includes the current username, the name of the computer, and the current directory. Thus in the above picture, the user is "sandbox", the computer is "laptop", and the directory is "~", which is actually short for the home directory.

Entering commands

Throughout this document you will be entering many commands. Each time, type in the words and symbols as given and press Enter (or Return) at the end.

Keep in mind that, as with almost everything else in Linux, the commands are case-sensitive. So, if you're supposed to type in "ls", typing "LS" won't work.

Exiting from the terminal

The first thing you might want to know is how to exit. Simply type in "exit" and press Enter, or else press Ctrl-D. The terminal window will quit. Although it is also possible to exit by clicking the "close" button or menu-option, this is the preferred method since it ensures that you are not in a text editor or other application where you could lose data.

A simple command

In the terminal window, enter "ls" and press Enter. You should see something like this (the actual contents may vary):

```
sandbox@laptop:~ > ls
KDesktop  public_html  snapshot1.png
sandbox@laptop:~ >
```

The "ls" (list) command lists the contents of the current directory.

When used from a terminal, it generally uses colours to differentiate between directories, images, executable files etc. As you can see, the prompt reappears at the end.

Adding options

Like practically all commands in Linux, you can add options to the "ls" command to alter its output or influence its behaviour. An option is preceded by a dash (eg, "ls -a ") Try out the following variations of the ls command, to see different forms of output:

```
ls -l
```

Produces a "long format" directory listing. For each file or directory, it also shows the owner, group, size, date modified and permissions

```
ls -a
```

Lists all the files in the directory, including hidden ones. In Linux, files that start with a period (.) are usually not shown.

```
ls -R
```

Lists the contents of each subdirectory, their subdirectories etc (recursive).

When you want to give more than one option, you can group them together with a single dash. For example, the command "ls -al " is the same as "ls -a -l"

Some options consist of a word (or words) instead of a letter, and have two dashes instead of one. For example, the command "ls -l --full-time" displays the date and time of modification in full.

Finally, some options may also have a value. For example, "ls -l --sort=size " sorts the listing by size.

Adding parameters

Apart from options (which are preceded by one or two dashes), you can also specify parameters, such as filenames, directory names and so on.

For example with the "ls" command, if you don't specify any parameter, it will list the contents of the current directory. However, you could instead give it a parameter specifying what to list. For example if you type in "ls /usr", it will list the contents of the "/usr" directory.

You can specify more than one parameter, but we'll see more about that later.

Obtaining help

Contrary to popular belief, commands and programs in Linux tend to be very well documented – usually more so than in Windows, which tends to document

commands only if they are very popular, and then only document up to a certain "level".

This section outlines the main ways of getting such information. Note that many distributions, such as Mandrake, RedHat and SuSE, have provided a graphical user interface which can be used to access such information using something like a specialised browser.

The "man" command

Almost every command in Linux has online help available from the command line, through the "man" (manual) command.

Try it now – type in "man ls". The resulting page will describe the command, then describe every option, then give further details about the program, the author, and so on. This information is shown using the "less" command (which we'll describe later on). For now, it issufficient to know that you can use the up and down arrow, PgUp and PgDn keys to move around, and the Q key to quit.

The "info" command

Another source of online help is the "info" command. Some Linux commands may supply both "man" and "info" documentation. As a general rule, "info" documentation is more verbose and descriptive, like a user guide, while "man" documentation is more like a reference manual, giving lists of options and parameters, and the meaning of each.

Try typing "info ls" now. The method for moving around in "info" is quite similar to "man" – you can also use the arrows and PgUp/PgDn to move, and Q to quit. The main difference is that info pages can contain "menus" of links which lead to other pages. To follow a link, move the text cursor to it with the arrow keys, and press Enter.

The "--help" option

Most (but not all) programs have a --help option which displays a very short description of its main options and parameters. Try typing "ls --help" to see. This will produce more than one screenful of information, so you'll have to use the terminal's scrollbar to see what was displayed.

The "--help" information rarely says anything that isn't also found in the "man" documentation, so it's rarely needed, except in a tiny number of programs which do not supply any other form of documentation.

The Linux Documentation Project

The L.D.P. is a project which collects not only all the man and info pages, but has a huge collection of longer guides, howtos and mini-howtos on a wide variety of topics. Unlike man and info pages, these howtos are not about a specific command, but rather how to accomplish a particular task. When you're looking for information on the internet, this is probably the best place to start.

Linux Directories

As you probably already know from working in graphics mode, in Linux the directories (aka "folders") use the slash (/) as a separator (Windows uses backslash (\)). In other words it works just like websites or ftp servers.

Any directory which starts with a slash, such as "/usr/bin", means it is an "absolute" name – the name specifies the entire sequence of directories from the "root" directory (/) up to the specific directory being requested (bin). Thus, it doesn't matter which directory is the "current" directory when you specify that name, it will always point to the /usr/bin directory.

On the other hand a directory which does not start with a slash is relative to the current directory. For example the directory "bin" will point to different directories depending on whether you are in the root directory (in which case it will point to "/bin"), in the "/usr" directory (in which case it will point to /usr/bin) or in the "/usr/local" directory (in which case it will point to /usr/local/bin).

The same applies to files – if you specify "file.txt" it is assumed to be in the current directory, while if you specify "/tmp/file.txt" it will always point to "file.txt" in the temporary directory.

Two special directory names are the current directory, represented by a single period (.) and the parent directory, represented by a double period (..). Thus, if you are in the /home/sandbox directory and type in ls .., it will list the contents of the parent directory, which is /home.

Some System Directories

Below is a list of some common directories that are found in Linux and Unix systems, and what they are used for.

/

This is the root directory, inside which all other directories reside This is similar to the root directory of a drive in Windows (C:\), except that in Linux even different hard disks reside within this root.

/bin

This stands for "binary", and contains program (executable) files. This (and other "bin" directories) is where commands such as "ls" can be found.

In Windows, the c:\windows\commands holds some of command-line programs, but others are scattered in various other directories.

/dev

This stands for "devices". It contains a number of special pseudo-files that are used to access the physical hardware that make up, or are connected to, your computer. For example the parallel-port would be a file called "lp0" in this directory, while the hard disk would be "hda", and its first partition would be "hda0".

Windows/DOS uses a similar method, however in Windows these are not in any particular directory. Devices have names like LPT1, COM1 or CON -any time you try to access a file with that name from any directory, you will get the parallel printer, serial port or console, respectively.

/etc

This is where (almost) all system-wide configuration information is stored. Almost all configuration information is stored in text files, so you can go into this directory and have a look around with a text viewer if you like. Some of the files are quite cryptic though.

There is no equivalent in Windows, where configuration data can be stored anywhere, including the registry, INI files and other data files in various directories.

/home

This is where users' home directories are usually found. Thus, if you created a user called "sandbox", there will be a directory with the same name in this directory, which will be that user's home directory.

The nearest equivalent in Windows is c:\windows\profiles, where some user-specific data is held, together with c:\My Documents, where user-created documents go. However other data can be written in many other directories.

/lib

This is where the library files are found. Libraries are files containing reusable functions and routines for programs to use.

There is no equivalent in Windows/DOS.

/mnt

This is where storage devices other than the hard disks are mounted.

This directory usually contains subdirectories called "cdrom", "floppy", etc., which - when these devices are mounted - show the contents of the CD-ROM or floppy disk respectively. Your Windows drives may also be automatically mounted in this directory.
There is no equivalent in Windows/DOS.

/OPT

This is where optional components of the system are installed. Products such as KDE, Gnome and Oracle may be installed into this directory.

The nearest thing in Windows is the c:\Program Files directory.

/TMP

This is a temporary directory. All files placed in here will automatically be deleted eventually.

The equivalent in Windows/DOS is c:\windows\temp.

/USR

Contains a copy of most of the directories in the root. For example, there is a "bin" directory containing programs, a "lib" directory containing libraries, etc. Usually, "core" Linux files are contained in the root direcories, while "non-core" files are in the "/usr" subdirectories.

There is no equivalent in Windows/DOS.

/VAR

Stands for "various". Among the files stored here are the system log files, spool files and other data files.

There is no equivalent in Windows/DOS.

DIRECTORY COMMANDS

Here are the most common commands to work with directories.

`mkdir new-directory-name`

Creates a new directory, "new-directory-name"

`cd directory-name`

Goes to the specified directory, making it the "current directory"

`cd`

When you don't give a directory name, goes to your "home" directory.

`rmdir directory-name`

Removes (deletes) the directory. As a safety measure, the directory must be empty before it can be deleted.

`pwd`

Displays the current directory.

`ls directory-name`

Lists the contents of the directory.

The following sequence of commands (and results) demonstrates the above commands. For clarity, the prompt has been coloured grey. After displaying the directory contents, a new directory called "testing" is created and the directory listed again. Then we go into the new directory, display the current directory, go back to the "home" directory, and display the current directory again. Finally the "testing" directory is removed.

```
sandbox@laptop:~ > ls
KDesktop  public_html
sandbox@laptop:~ > mkdir testing
sandbox@laptop:~ > ls
KDesktop  public_html  testing
sandbox@laptop:~ > cd testing
sandbox@laptop:~/testing > pwd
/home/sandbox/testing
sandbox@laptop:~/testing > cd
sandbox@laptop:~ > pwd
/home/sandbox
sandbox@laptop:~ > rmdir testing
sandbox@laptop:~ >
```

Linux Files

File Commands

Here are some of the common commands to work with files.

```
cp filename1 filename2
cp filename1 filename2 filename2 (etc) directory
```

Copies a file, from filename1 to filename2 or (second form) copies one or more files into the specified directory. Warning: if the destination file already exists, it will be overwritten.

```
mv filename1 filename2
```

Renames a file, from filename1 to filename2. Warning: if the second file already exists, it will be overwritten.

```
mv filename1 filename2 filename2 (etc) directory
```

Moves one or more files into the specified directory. Warning: if the directory already contains files with the same names, they will be overwritten.

```
less filename
```

Displays the contents of the specified file onto the screen, allowing you to use the arrow keys, PgUp/PgDown etc to move around (like the "man" command).

```
file filename
```

Displays the file-type by examining its contents, with a very high degree of accuracy.

```
locate file-or-directory-name
```

searches for a file or directory in the entire hard disk and displays all the places it's found. You can also specify a partial name or a section of the entire path.

Wildcards

Wherever you can specify a file or directory name in Linux, you can use wildcards. By using one or more special symbols, the shell will find those files which match a pattern, and place them on the command line instead of the pattern itself. The word "wild card" refers to the "Joker" in a pack of cards, since this card can stand for any other card in many card games. In the same way, the "wildcard" character can stand for other letters and characters in a filename.

Testing Wildcards

To get the hang of wildcards, the best thing to do is to go to a directory which is full of files and try using the "ls" command with the wildcards as arguments. As we saw before, the "ls" command can take a parameter which tells it what to display. Instead of giving it a directory, we're going to pass it a list of all filenames to display.

This list will come from the wildcard patterns which we will see below.

So, before you continue, in the terminal window type the command "cd /usr/bin". This will switch to the main directory containing the operating system commands. It's full of files, so it's ideal for our experiments.

```
sandbox@laptop:~> cd /usr/bin
sandbox@laptop:/usr/bin>
```

*The * wildcard*

The first wildcard is the asterisk (*). The asterisk stands for zero or more other characters. By placing this wildcard at the beginning, middle or end of a pattern, you can build a pattern which has the rest of the pattern at one or either end. For example the pattern "*txt" means any sequence of letters which ends with "txt".

Below is a table of patterns, and an example of which filenames would match such a pattern, and others that would NOT match.

Pattern	Matching files	NON-matching files	Why not
*.txt	File.txtanother-file.txttxt	File.TXTFile.txt 2txtfileFiletxt	TXT is uppercase ends in "2""txt" is not at the end no period (.)
*txt	File.txtanother-file.txttxtFiletxt	File.TXTFile.txt 2txtfile	TXT is uppercase ends in "2""txt" is not at the end
txt	File.txtanother-file.txttxtFiletxtFile.txt 2txtfile	File.TXT	uppercase

The ? wildcard

While the * wildcard could stand for zero or more letters or characters, the ? wildcard stands for exactly one. Thus, a pattern of "???" stands for filenames which are exactly three characters long. The pattern

"x??" matches any three-letter filename which starts with "x".

The [] wildcard

The square brackets are used to contain a set of characters to match.

For example, the pattern "[ABC]*" matches any filename which starts in one of the letters A B or C, followed by zero or more characters.

If the first character is an exclamation mark (!) or caret (^), then the pattern matches any character except those given. Thus, the pattern "[^x]*" means any filename except those starting with "x".

Instead of individual letters, the set can contain a range. For example, the pattern "[A-Z]*" means any filename which starts with an uppercase letter between A and Z inclusive, followed by zero or more other characters, while "[A-Za-z123]" means a single character which is an uppercase or lowercase letter, or the digits 1, 2 or 3.

How wildcards work

There is a very significant difference between the way that Windows handles wildcards and how this is done in Linux or other Unixes. In Windows, the program or command being executed receives the wildcard expression intact. If a program was not designed to cater for wildcards, it will try to open a file called (say) "*.txt".

In Linux, on the other hand, it is the bash shell that does all the work. It takes the pattern containing wildcards, convert it into a list of matching filenames, and pass that to the program in place of the pattern. The table below shows how certain commands would be "translated" by bash. The actual filenames depend on the contents of the directory so they could vary.

Original command	*What actually gets executed*
ls	ls
ls y*	ls yacc ybmtopbm yes ypcat ypchfn ypchsh ypmatch yppasswd ypwhich yuvsplittoppm yuvtoppm
ls ?a?	ls cal man tac
ls blubble*	ls blubble*

Note the last example - bash could find no file with that name so the pattern, including wildcards, is passed to the program "as is".

This makes things easier for the programs themselves, because they only need to cater for lists of filenames on their command lines. However, there are a few techniques that one could do in MS-DOS that cannot be done in Linux. For example in MS-DOS, one could enter the command "copy *.doc *.bak" - which would copy all files ending with "doc" into an equivalent filename, but ending in "bak". In Linux, that command would be translated to something like "copy file1.doc file2.doc file3.doc file2.bak file4.bak" - giving a totally different and probably undesired result. Actually this technique no longer works well in Windows due to the introduction of long file names.

Wildcards with directories

Wildcards with Linux work on directories too. For example, the pattern "*/file.txt" means, all files called "file.txt" in any subdirectory.

Hidden files

Wildcards will not match hidden files unless the wildcard pattern itself starts with a period. Thus, the pattern ".*" matches all hidden files (hidden files are files which start with a period, such as .profile or .kde2)

Typing Tricks

When you're in the bash prompt, you can use the up- and down-arrow keys to recall previously typed commands.

You can also press Ctrl-R and start typing part of another command to find the last command that contains the letters you are typing. Thus if you want to find the last change-directory, type "[Ctrl-R]cd", and the command line will display the last "cd" command you typed.

If you start typing a filename or directory name, you can press [Tab] and bash will complete the file or directory name for you, assuming that such a file exists and is the only one that starts with the typed-in part. For example, if you type "ls br[Tab]", bash will complete the filename to "brushtopbm", if this file exists and is the only file starting with "br".

Redirecting Output

Redirecting output to a file

Most programs, when executed, display lots of text on the screen. You can save this text into a file if you want to retain it, or process it further. To do this, you use the redirection operator, ">".

For example, suppose that you want to save a copy of a directory listing into a file. You would type in the complete "ls" command, followed by ">" and the name of the new file to be created.

In the following example, we will list the contents of the /usr directory in long format, and write this listing into a file in our home directory (since we cannot create files in the /usr directory).

```
sandbox@laptop:~> cd /usr
sandbox@laptop:/usr> ls -l > ~/usr-listing.txt
sandbox@laptop:/usr> cd
sandbox@laptop:~> ls
```

```
KDesktop  public_html  snapshot1.png  usr-
listing.txt

sandbox@laptop:~>
```

Note the second line. First we have the command to list the directory "ls -l", then we have the redirection symbol ">" telling bash to dump all the results into a file, and finally we have the name of the new file to create (~/usr-listing.txt). As you may recall, the "~" symbol stands for the user's home directory.

The symbol we gave so far (>) creates a new file with the specified name. If that file already exists it is first emptied before it receives the program's output. If you give a double-angle-bracket (>>), the output is appended to the end of the file if it already exists. This can be used to redirect the output of several commands into one file.

Piping output to a program

When we used redirection, above, the output of a command was sent into a new file. With pipes, this output of one program is instead sent to the input of another program. This second program processes the output of the first program, and may produce its own output based on it.

Let us try this out. The "cat" command displays the contents of a file. The "/etc/services" file contains a list of recognised TCP/IP services.

Try displaying this file now - it should show you many screenfuls of text.

Note that in the following example only a small number of lines are shown: your system will have much more.

```
sandbox@laptop:~ >cat /etc/services
#                 0/tcp    Reserved
#                 0/udp    Reserved
tcpmux            1/tcp                  # TCP Port Service
Multiplexer
tcpmux            1/udp                  # TCP Port Service
Multiplexer
compressnet       2/tcp                  # Management Utility
compressnet       2/udp                  # Management Utility
......lots more lines.....
nimhub          48002/tcp                # Nimbus Hub
nimhub          48002/udp                # Nimbus Hub
nimgtw          48003/tcp                # Nimbus Gateway
nimgtw          48003/udp                # Nimbus Gateway
```

Now, the "sort" command (as its name implies) sorts its input in alphabetical order, and sends it to the output. So, in order to sort the output of the "cat" command, we must pipe it to the sort command. To do this, we use the pipe symbol (|).

```
sandbox@laptop:~ >cat /etc/services | sort
3Com-nsd        1742/tcp                 # 3Com-nsd
3Com-nsd        1742/udp                 # 3Com-nsd
3com-amp3       629/tcp                  # 3Com AMP3
3com-amp3       629/udp                  # 3Com AMP3
......lots more lines.....
zip             6/ddp                    # Zone Information Protocol
zserv           346/tcp                  # Zebra server
zserv           346/udp                  # Zebra server
```

One very popular use of the pipe is to send the output of such long commands to the "less" command. This command allows you to scroll up and down, or even sideways, to view the complete results.

```
ps -Hefw | less
```

The above command line displays the list of processes in the system and uses "less" to allow you to scroll around (using the arrow keys and PgUp/PgDown), and Q to quit.

It is possible to string together several commands separated by a pipe. For example we could use the "cat" command to display the contents of the services file above, pipe that to the "sort" command to sort it, then pipe the results of the sort command into the "tail" command to pick out only the last 50 lines, and finally pipe those 50 lines to the "less" command to scroll around in the results.

```
cat /etc/services | sort | tail -n 50 | less
```

Environment variables

In Linux (as well as MS-DOS, Windows etc), each program has a number of "environment variables". These are a series of "settings" which can be used to control certain aspects of the system, user preferences, etc.

Whenever a program is executed it receives a copy of the environment variables from the process which executes it. So, if you run a program from the bash shell, that program will receive a copy of all the environment variables from the bash shell itself.

In the following example, I am using the "date" program to display the date. However, I am changing the language-code each time. The "export" command is used to set a new value in an environment variable, or to create it if it doesn't exist.

```
sandbox@laptop:~>date
Thu Aug  2 04:35:55 CEST 2001
sandbox@laptop:~>export LANG=it_IT
sandbox@laptop:~>date
gio ago  2 04:45:05 CEST 2001
sandbox@laptop:~>export LANG=mt_MT
sandbox@laptop:~>date
?am Awi  2 04:45:55 CEST 2001
```

You can use the "set" command to view all your current environment variables. Since there is usually more than one screenful, you might want to pipe that to the "less" command to view it one screen at a time.

You can include the value of any environment variable within a command by preceding it with a dollar sign. For example, the command "ls -l $SHELL" will produce a long-listing of the bash program itself, since the SHELL environment variable contains the location of the current shell.

Among the most important environment variables are HOME, PATH and PS1.

The first obviously points to the user's home directory. It can be used to construct filenames, such as "$HOME/file1.txt".

The PATH

The second variable, PATH, is a list of directory

names separated by colons. This is the list of places that bash will look in when searching for the command names you specify. Thus, when you type "ls", bash will start searching in these directories one at a time until it finds an executable file called "ls", and executes it.

You can use "$PATH" within another command to set the PATH to a different value, in order to add a new directory to the PATH. For example, suppose that you create a new directory for programs in your own home directory. You can add this new directory to the PATH so that bash will look inside it too. To do that (assuming your new directory is called "bin"), you could give the following command:

```
export PATH=$PATH:$HOME/bin
```

In that command, I am setting the PATH environment variable to the previous value of itself, followed by a colon, and a directory-name formed from the HOME variable and "/bin".

The PS1 prompt

The last environment variables, PS1, tells bash what to display in the prompt. You can play around with the PS1 variable to produce a prompt that you like. While producing this document, for example, I set up the prompt to contain the required HTML code to colour itself grey.

```
sandbox@laptop:~ >export PS1="I am \u on \h> "
I am sandbox on laptop>
```

Note however than any changes you make to the environment variables will be temporary. They will revert to normal next time you open a terminal window, or log on. We will see how to make these settings permanent in a moment.

Scripts

A shell script is a series of shell commands, similar to the ones given above, contained in a file, which can be given to the bash program to be executed in sequence. Technically, it is a program with "bash" as the language.

You can use any text editor to create your shell script - simply choose one you are comfortable with. If you are using KDE you can use the Advanced Editor (kate or kwrite), or kedit. Do not use a word processing program - these are not text editors, and insert additional formatting information into the text. If, however, you know how to use your word processor to save plain-text file, go ahead and use it.

Start with a new file, and type the following lines.

```
#!/bin/bash
echo "I am about to list the home directory"
# here is the actual listing:
ls $HOME
echo "Done!"
```

The first line has a special meaning. It tells the current shell which program should be used to interpret this file. Here, we have given the filename of the bash program. This is so that, if this script is invoked from within a different shell, or from a file-management program such as Konqueror and Nautilus, it will still know that this script requires the bash shell to run it.

The second line is our first command. We use the "echo" command to display a simple line of information.

The third line is a comment. Bash simply ignores it, but it can be useful to a person reading the program in the future to understand what the code is trying to do. While comments are not very useful in such a short, simple script, they are vital in longer and more cryptic ones.

This is followed by two more commands - the "ls" command which takes the directory name as a parameter, and finally another "echo" to display our final declaration of triumph!

When you save this file, it shows up as an ordinary file which cannot be executed. In order to run it, you must make it executable. To do this, you use the "chmod" command. In the following example, I have used "myscript" as the script name - you should change that to whichever name you used.

```
chmod u+x myscript
```

That's it! You can now execute the new script. To do this, enter the folloowing command, again replacing "myscript" with the name you used if different.

```
sandbox@laptop:~ >./myscript
```

```
I am about to list the home directory
KDesktop  myscript  public_html  snapshot1.png
usr-listing.txt
Done!
```

Note the "./" at the beginning of the file - this is actually a directory name (a single period means the current directory). You can avoid this by placing the new script, and any others you create, in a directory within the PATH environment variable. Then you can enter the command from wherever you are.

Bash's startup scripts

Bash uses a special, hidden file in your home directory called ".bashrc". This is essentially the same as any other script, except that it lacks the first line we used above, and it is executed automatically by bash as soon as it starts up. This is where you should place any modified environment variables or other settings for them to be "remembered" by bash every time you log in.

Aliases

Bash allows you to define new commands which translate to other, longer commands. For example, if you type the following...

```
alias lh="ls -l -a $HOME"
```

, then the command "lh" will become equivalent to the longer command "lh -l -a $HOME".

You can even use this to re-define existing commands. For example if you type "alias ls="ls -a"", then the ls command will always start showing hidden files. You can remove an existing alias by using the unalias command.

You can place these aliases into the .bashrc file as well, to be set up when you load bash.

You can even add "safety features" to certain commands. For example, if you create an alias which makes the copy (cp) command equal to "cp -i", then the copy command will prompt you every time you are about to copy onto an existing file, instead of overwriting it without warning.

However you have to keep in mind that these settings will not exist when you're working in a different user, until you add them to the .bashrc file of that user as well.

Switching to root

Occasionally you may encounter a situation where you have to switch to the root user to perform a certain command. Before doing so you should be aware that, while working as the root user you are able to do anything. You can install files to the system areas, or delete all the data on your hard disk. Be careful what you do while logged in as root.

The command to switch from one user to another is su - which stands for "set user". When you enter this command without any parameters it will switch to the root user. Alternatively, you can follow it with the username to which you want to switch. In either case you will be prompted for the password for that user.

When you enter the password you will be working as that user - you will be able to access the files that that user can access, perform tasks he is entitled to perform, etc. In the case of the root user, that means all files and all tasks.

Try it first with your own normal username. In this example, I am using 'ramon' as the other username.

```
ramon@laptop:/home/sandbox> ls /home/ramon
Access denied.
ramon@laptop:/home/sandbox> ls
file1.txt   file2.txt
sandbox@laptop:~> su ramon
Password: (enter ramon's password)
ramon@laptop:/home/sandbox> ls
Access denied.
ramon@laptop:/home/sandbox> ls /home/ramon
Desktop  Mail  Documents
ramon@laptop:/home/sandbox> exit
sandbox@laptop:~>
```

Note that, while logged in as "sandbox" I could list the contents of sandbox's home directory but was denied access to ramon's home directory, and vice versa. Note also that I used the exit command to exit from the "ramon" user back to the sandbox user. This is because the su command creates a new shell process using the "ramon" user.

If we use the su command without parameters we are prompted for root's password. Once logged in as root we can access all areas.

```
sandbox@laptop:~> su
Password: (enter root's password)
root@laptop:/home/sandbox> ls
file1.txt   file2.txt
root@laptop:/home/sandbox> ls /home/ramon
Desktop  Mail  Documents
root@laptop:/home/sandbox> exit
sandbox@laptop:~>
```

There is a difference between logging in as root (or any user) then loading the terminal, and starting off as another user, loading the terminal and then using su to switch to root. This difference lies in the startup-scripts that get loaded. When you use the "su" command as shown above, it inherits all the settings from the previous shell. Thus, for instance, if you have set up some environment variables they will be carried over and are still accessible as root. This is useful in some compilations. If you use your regular user to compile a program and then switch to root to perform the final installation step, it may be useful to carry over any environment variables you may have set up.

However this has the disadvantage that any commands in root's startup scripts will not get executed. If you ever need to switch to root (or some other user) and ensure that the startup scripts DO get executed, add a hyphen (-) between the su command and the username parameter, if any. Thus, the commands used above would become:

```
su - ramon
su -
```

You will still be prompted for the password, but this time the startup scripts will be executed and any settings from your own shell session will be forgotten - it will be as if you had logged in as that user in the login screen.

Compiling from source

One of the main reasons why some people may need to use a terminal window is to compile a program which they downloaded from the web.

Compiling is a process that takes text source code and converts it into a binary program, which can only run on the specific processor where it was compiled.

All such programs include intructions on how to compile them - these instructions are found in text files called "INSTALL", "README" or some similar name. Here I will describe one of the commonest routines used.

First of all, assume that the file we downloaded was called "MyProgram-1.2.3.tgz". Since this file is an archive, like a WinZip file, we have to extract its contents. This is done using the tar command:

```
tar -xvzf MyProgram-1.2.3.tgz
```

The options given instruct tar to extract (x) and verbose (v) the zipped contents (z) of an archive file (f) with the specified name. Tar will extract the contents of the archive and display what it is doing.

Next, we have to cd to the newly created directory, which generally has the same name as the archive, minus the ".tgz".

```
cd MyProgram-1.2.3
```

Usually, the next step is to run the "configure" script. This examines your system and produces a compilation and installation script that is ideally suited for your configuration. Another task done at this stage is to check whether you have all prerequisite software installed. The "configure" command produces many lines of output. Unless it ends with an error, these can be ignored.

./configure

When the "configure" script writes the recipe, the "make" command takes the ingredients and bakes the cake. This is often done in two steps.

First, you type in "make" to generate the binary program, then you type "make install" to place these binary programs in their proper location on your computer. This last step must be performed using the "root" user, since it requires permission to place files in system areas. You can switch to the root user by using the "su" command.

```
make
su
make install
exit
```

If no errors were reported, you're ready.

Conclusion

That concludes our basic overview of how to use a command shell. There is a lot more, which would be very useful especially when writing shell scripts or performing advanced tasks. However the above should be serve as a good beginning to understand what is going on in the system, as well as a starting point from which you can learn more details about Linux shell programs. Remember that the bash shell has its own man page, as well as longer tutorials and documentation available at the Linux Documentation Project website.

Comments
If you would like to send comments, suggestions etc. please send them to me at ramon.casha@linux.org.mt, especially if you found any section hard to understand.

This article is re-printed with permission. The originals can be found at:
http://linux.org.mt/article/terminal

Samba 2.2.2 and Mandrake Linux

Author: Buchan Milne <buchanmilne@netscape.net>

For those of you who don't know yet, Samba is the software used on unix systems to provide windows networking services (file and print sharing). I am sure many Mandrake users already use samba at home to access files on their windows boxes, and as shown in a previous article, samba can replace a Widows Domain Controller.

This article covers some of the new features made availabe in samba since samba-2.0.x, which is what shipped with Mandrake 8.0. Please note that quite a bit of effort has gone into providing sample comfigurations in the smb.conf file. In most cases, you can implement almost any setup of samba just by uncommenting sections of /etc/samba/smb.conf

Since the release of Mandrake 8.0, a lot has happened on the Samba side:

Samba 2.2.0 release
The release of samba 2.2.0 ocurred shortly after the release of Mandrake 8.0, and was the first version to allow WIndows 2000 machines to join a Samba controlled domain. Point-and-print (a feature available since Windows NT4) was also added. Unfortunately there were some limitations, for example Windows NT 4 workstations could not be used for file-sharing in a samba 2.2.2 domain.

Samba 2.2.1 release
The samba 2.2.1 release fixed the problem with sharing between Windows NT4 machines, and allowed Windows 2000 SP2 machines to be joined to a samba domain. nss_wins was also released, which allows name resolution via netbios names

Samba 2.2.2 release
The samba 2.2.2 release is significant, since it allows Windows XP clients to be joined to a windows domain, and also now includes winbind, which allows unix/samba machines to better integrate into a Windows controlled domain.

POINT-AND PRINT

Point-and-print means that you can now install the print drivers on the print server, and the clients will automatically download and install the drivers on the client machines. This is useful in a secure Windows NT/Windows 2000 network, as it allows users to install print drivers (which would normally require a user with admin rights to log in and install the printer). To use this, you need to enable the "print$" share in the smb.conf file provided by the samba RPM. The samba-howto-collection is also reccomended reading in this regard. Look for some more documentation on Mandrakeuser.org about this in the next few weeks.

Windows 2000 and Windows XP domain members

The details of joining Windows XP and Windows 2000 clients to a samba domain are beyond the scope of this post, but you can see detailed instructions on http://MandrakeUser.org/

Winbind

Winbind is a daemon and two shared libraries which allow you to do all user/group enumeration and user authentication from a Windows Domain. Previously, you were able to join a samba box to a windows domain, and do all samba authentication via the domain, however this required that unix user accounts be created on the samba box (or via some other method such as NIS or LDAP). Now, once winbind is running, all services which are pam-enabled (including uw-imap, apache with mod_auth_external, ssh, cups, kde, gdm etc) can do authentication from the windows domain. This means that a linux-mandrake box running winbind can be joined to a windows domain and act (in most regards) like a Windows NT/Windows 2000/Windows XP domain member.

Details on how to set winbind up are available at MandrakeUser.org

XFS ACLs

As you may have seen in a previous post, XFS-ACLs have also been available since samba-2.2.1a (and work out the box on Mandrake 8.1). This means that you can manipulate ACLs (access contol lists) on a Mandrake 8.1/samba box from the security tab on the file properties window on windows machines. For those of you who might have Mandrake 8.0 boxes out there, I have made an XFS-enable kernel and associated user-space tools, and Mandrake 2.2.1a compiled with ACLs available here. Samba 2.2.2 RPMs for Mandrake 8.0 with ACL support will be available as soon as I can get to a Mandrake 8.0 box with a compiler (maybe tomorrow) from here.

nss_wins

nss_wins is another shared library, which allows name resolution via netbios names. What does this mean? It means, that finally, you can do:

```
# ping windows_machine
```

where windows_machine is the netbios name (as set under the network control panel applet on the identification tab). To enable this, you need to add a "wins" entry to /etc/nsswitch.conf. In the samba-2.2.2. RPMs this is done for you.

Packaging changes

Since Mandrake 8.0, the following packaging changes have been made:

- Samba configuration files have been moved to /etc/samba
- Documentation has become so large, is it now a sub-package: samba-doc
- Encrypted passwords are enabled by default, meaning you need to create a smbpasswd for users to access your samba box, via sambpasswd -a username. You can turn this off in /etc/samba/smb.conf
- New package has been added for nss_wins
- New package added for winbind, and a winbind start-up script

New features not enabled

Samba-2.2.2 has abstracted the password management, and will allow smbpasswd's to be stored in LDAP, NIS+ or a TDB (trivial database). These are experimental features, but some people are using them effectively. Note that LDAP+Samba can provide a setup similar to Active Directory, allowing you to have multiple samba domains with consistent usernames and passwords.

Where to get samba-2.2.2

AFAIK Samba 2.2.2 RPMs should be available via Security Updates as soon as the QA team is finished, but they will not provide winbind or nss_wins. For those of you brave enough to use these new features, you should get RPMs in cooker, or RPMs compiled for Mandrake 8.1 here or here.

What can I do to help?

We haven't been able to fully test winbind, so any of you who have the opportunity to use or test winbind in a Windows Domain will be able to help just by letting us know if it worked. Civileme has mentioned that a sub-group for crashtesting samba might be on the crashtesters list, so if you have things you want to test, you can join us there. If you have troubles or would like to help more, let me (bgmilne at cae dot co dot za) know.

What qualifies me to tell you about all this new stuff in samba?

I set up a samba-controlled network with samba-2.0.6 on Mandrake 7.1 about 15 months ago, and have been needing some of the advanced features in samba (especially WIndows 2000 domain members and ACLs), and so have been involved in packaging samba for Mandrake. Our domain controller currently runs samba-2.2.1a on Mandrake 8.0, although I

would like to use XFS-ACLs on it, so will put on the XFS-enabled kernel and samba-2.2.2 as soon as we can afford to take the machine down. We have a Windows 2000 member server running MSSQL 7 (not by choice). We have about 60 desktop machines, mostly Windows 2000, with a few NT4 boxes left. We also run a few Mandrake 8.x desktop machines, with user enumeration via LDAP and authentication via pam_smb, allowing users to have one password for windows and linux (pam_smb), and consistent uids so we can use NFS (via LDAP). Storing samba passwords in LDAP is the next step, so we can keep accounts synchronised at our remote facility.

I have also been doing a bit of rpm packaging, for example:

- pam_smb (http://www.rpmhelp.net/modules.php?op=modload&name=NS-Traktopel_RPMS&file=index&rid=93),as mentioned above
- rdesktop (http://www.rpmhelp.net/modules.php?op=modload&name=NS-Traktopel_RPMS&file=index&rid=95), a windows terminal service client, which allows fast desktop access to our Windows 2000 Server
- and quite a few other useful RPMs that you can find at http://ranger.dnsalias.com/mandrake

This article is re-printed with permission. The originals can be found at:
http://ranger.dnsalias.com
(and http://ranger.dnsalias.com/mandrake/samba for RPMs and more docs)

Encryption for the masses

Author: Glenn Mullikin <glmull@machineofthemonth.org>

We hear alot about security on the internet, about securing your system from hackers who don't have much reason for existence except to break into peoples' systems. Maybe that's a good existence because they can get a book deal but not for you. But what happens when they infiltrate your front lines?

What if they get access to your files? Then it might be all over. Unless you have a second line of defense. Introducing public-key cryptography...

GnuPG works

Typically, your most important files are the ones that you don't want anyone to see. But what if you want or need to keep them on that production machine? Well, maybe you can do it in encrypted form. And GnuPG is good for encrypting stuff into something no one can make sense of except the person in possession of the private key. Notice I said person and not persons.

If someone gets your private key then they can use it to unlock all your encrypted files, assuming they were able to deal with the passphrase issue. That's right. So don't let them do that. I haven't been using the product that long to know the type of techniques this would entail although I suspect you would want to move your keyring to a removeable storeage medium or put the keyring on a computer you considered to be very secure. And then create a symbolic link to your real keyring from /home/username/.gnupg, which is the directory where the keys are stored by default. But anyway, yes this is something to consider because your private key is very important to keep secret.

Let's take a look at my keyring permissions:

```
[Wed May 31 12:00:35][glenn_m localhost][jobs 1][~]$ ls -al .gnupg
total 15
drwx--S---    2 glenn_m  glenn_m      1024 May 30 11:16 .
drwxr-sr-x   54 glenn_m  glenn_m      5120 May 31 11:14 ..
-rw-rw-r--    1 glenn_m  glenn_m      2924 Mar 25 17:07 options
-rw-rw-r--    1 glenn_m  glenn_m       900 Mar 25 17:11 pubring.gpg
-rw-rw-r--    1 glenn_m  glenn_m         0 Mar 25 17:07
pubring.gpg~
-rw-------    1 glenn_m  glenn_m      1218 Mar 25 17:11 secring.gpg
-rw-rw-r--    1 glenn_m  glenn_m      2560 Mar 25
17:18 trustdb.gpg
[Wed May 31 12:00:42][glenn_m localhost][jobs 1][~]$
```

As you can see, secring.gpg is not world readable. And that is good. But the other files are world readable but I guess this is by design and not really a security issue.

Let's take a look at my public key:

```
[Wed May 31 12:02:34][glenn_m localhost][jobs 1][~]$ gpg --list-
keys
/home/glenn_m/.gnupg/pubring.gpg
--------------------------------
pub  1024D/91326DD5 2000-03-25 Glenn Mullikin (hello)
<glmull@altavista.com>
sub  1024g/8FEFBC6B 2000-03-25 [Wed May 31 12:03:34][glenn_m
localhost][jobs 1][~]$
```

It goes by the name of hello or glenn or glmull or whatever substring I want to use of "Glenn Mullikin (hello) <glmull@altavista.com>". Cool with me.

Let's say there is someone I wanted to send an encrypted file to. Well, I need their public key so I can encrypt that file using their public key. I'll leave that as an exercise. It's not difficult, it just that I haven't used it for that purpose. Then what do I use it for?

Me? For my own encryption needs. Let's take an example. Let's say I have a file called ccards, which I do have, and this file contains very important information relating to my credit cards. I don't want you or any unauthorized person to see the information in this file because it has my credit card numbers and pin number for each card. I also want to make sure I don't lose this information because if I did, well, it wouldn't be pretty. This suggests two things I might want to do. Backing up and encrypting. Each has a specific purpose and one does not erase the need for the other. In fact, you can integrate gnupg into your own backup shell script or perl script.

But back to our ccard file and how to encrypt it. Here is what I do:

```
[Wed May 31 12:10:57][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--armor --sign --output ccards.asc --encrypt ../../ccards You need
a passphrase to unlock the secret key for
```

```
user: "Glenn Mullikin (hello) <glmull@altavista.com>"
1024-bit DSA key, ID 91326DD5, created 2000-03-25 You did not
specify a user ID. (you may use "-r") Enter the user ID: glmull
[Wed May 31 12:12:00][glenn_m localhost][jobs 1][~/1test/gpg]$
```

Now let's disect the above command to see what's going on and what input was required to stdin after the command was run.

The --armor option produces an ASCII text file which is nice because then you can cut and paste into emails and other types of applications. Without the --armor option, you get some weird binary file the documentation says. And so just using the above --armor option means you can make GnuPG work with any email program. Just do the cust and paste thing. What about the --sign option? Well, it wasn't absolutely necessary and one could argue that if one is just encrypting files to onesself, it isn't necessary to perform authentication because presumably no one else could have created the ccard.asc file, right?

But in general, the --sign option performs a "hash" on the ccard file (the input file) and then uses the private key of the user to encrypt that hash. The recipient (which was prompted for by "Enter the user ID:" and I entered "glmull") will then be able to verify by using the sender's public key that the message is authentic, that is, that whoever sent the message is in possession of the private key of the supposed sender's public/private key pair. This doesn't absolutely prove that it is the person you think it is. This rests on how much faith you have in the supposed identity of the owner of the public key. I guess there are certificate authorities that can help with this such as verisign and even GnuPG has some facilities that allow you to assign trusts to your keys in your public key ring but ultimately, you are trusting some source of information. But anyway.

Let's say you called me on the telephone and said "I want to make sure I have the right public key for you." I would say "Well, what's the fingerprint?" And you would say "Oh, well, why don't you tell me what your fingerprint is?" And then I would say "Well, since I'm not an impostor and I want you to have the correct public key to make sure that someone else isn't masquerading as me, can you hold on for a second?" And you would say "Sure." And I would run the following command on my linux box:

```
[Wed May 31 12:12:00][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--fingerprint glmull
pub  1024D/91326DD5 2000-03-25 Glenn Mullikin (hello)
<glmull@altavista.com>
     Key fingerprint = 18D6 15C2 94DA 2A16 50A0  D760 78D0 8362
9132 6DD5
sub  1024g/8FEFBC6B 2000-03-25 [Wed May 31 12:23:57][glenn_m
localhost][jobs 1][~/1test/gpg]$
```

And then I would say, my "my key fingerprint is 18D6 15C2 94DA 2A16 50A0 D760 78D0 8362 9132 6DD5". That's alot better than saying "My public key is ..." and then reading it out because it could be very long and no one wants to do that. But you can publish your public key on the internet or anywhere else you wish so people can go get it. I will do that right now:

```
[Wed May 31 12:28:36][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--armor --export glmull >glennpubkey.asc
```

What does the "glennpubkey.asc" file contain? It contains the following:


```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org
mQGiBDjdOaMRBAC+S4Tfz9162yUP+kZDUSC/uILZ0C2DWf2grrRv/YA1vjLuVIN7
69iXQXv8Ee352Xnry8iOZKTSi0rqZSvqM/L69F0qnvR/SB53PVwoHBl4/hZTM4Wd
VcMMWGlHRSoSIoHZaKY3KgD0/pWZeprZdoia8LwXORtS4dgL19XSoTrHDwCghI5C
QT8b+IdoiIROwnaCWCveTBMD/1J9MZFCpMWbwg5S5zsT9kN4Cfgf0xISVya8ghrN
WSZ4kLvY4vSh9CsOCN6GO33AlOB0+bm3TFldnDbZpXiLU3nYvf6nrpvzUQbNKKRJ
F+787XTfaiPPnXxEvFaxZCaJwl9RgHEMZ8dq3yaHwVyONeYdnr+MzLd1r5U1p4E5
ZTIXA/9cx0i5z9DALCDQXGKZ9QpOnyhtELAkuFVMh60QngfPeShTKj6FxL69ZuTJ
JcVuqihX2Qt8yo+J2bIdkzLJ+LjlyeadXjAwQGp2MEl0UXFUyffNWSeLsFXFb9/D
iwtq88Aadwefin+jg63JeRy7cu1gmYrjg9ay27j/01E8QlSUOLQtR2xlbm4gTXVs
bGlraW4gKGhlbGxvKSA8Z2xtdWxsQGFsdGF2aXN0YS5jb20+iFYEExECABYFAjjd
OaMECwoEAwMVAwIDFgIBAheAAAoJEHjQg2KRMm3VbCgAnRn/uCpMv2BOtbBqNfme
kmDEEixsAJ0QV0D9iPOm2foGxJ7SgVriI1r1x7kBDQQ43TmrEAQAgD+e1zaWUw+7
0khLTAfPSKaPNAlqPcHWR8fdmNVRLaqs8K2CDmngMnBi8f246/K/cLLx3z/H8MiK
Nca9Hpmq01r2Ji9X3NLqL2n3MEDSB/nwaGY0NkhPxfDGdzVFMS4e8YOcqq8MB0Jt
GI8t89+tqiZ+LHQfvnAe51rN3bbG7psAAwYD/RQqbrC5r0tgRd/O9sWa6Mxm2lOP
SWRhxyu3Rwiqw4X+v58UafaW0eckpdgNRU9aRg70KaotBsV3JyfFUqsEU5j+bPUx
k/3VhxjgSOIr+73eUDcClfng7+jLJSxXn93xe4yx4Nl3MX4XOHFzSuaYPGF7QHIA
Q26D5wcppCveWV4hiEYEGBECAAYFAjjdOasACgkQeNCDYpEybdV4kwCfTNOyPBZp
F9CPptJ+/RhCOdD3/noAn2SVzkYfZxy+jKy3NUccEz+jCMDs
=byOq
-----END PGP PUBLIC KEY BLOCK-----
```


This is my public key, as long as a hacker doesn't get into my website and screw with it. :-)

And they both (fingerprint and actual key) give the same results as far as being able to tell whether or not the person has my public key or someone else's (that may be trying to masquerade as me!).

The_GNU_Privacy_Guard is a very powerful program. To appreciate it, you have to understand number theory and how the mathematics works to allow this type of secrecy. In fact, mathematics has become very important in the computer field these days.

But for most of us, the way in which the product works is a curiosity best left to the math gurus in ivory towers, however, the product is licensed under the GPL so its source is out in the open.

I want to go back and talk about encrypting again. In the example, I encryped my credit card text file and digitally signed it. Now what if I wanted to decrypt that file and check the digital signature? It's easy, let's do that now:

```
[Wed May 31 12:33:40][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--output ccards.dec --decrypt ccards.asc You need a passphrase to
unlock the secret key for
user: "Glenn Mullikin (hello) <glmull@altavista.com>"
1024-bit ELG-E key, ID 8FEFBC6B, created 2000-03-25 (main key ID
91326DD5) gpg: Signature made Wed May 31 12:12:00 2000 EDT using
DSA key ID 91326DD5
gpg: Good signature from "Glenn Mullikin (hello)
<glmull@altavista.com>"
[Wed May 31 12:35:10][glenn_m localhost][jobs 1][~/1test/gpg]$
```

There are a few things to notice above, that happened. The first thing is it prompted me for a passphrase. And that is because it will always do that since in order to decrypt something it has to access the secret key of the public/secret key pair. (The public key, being the key used to perform the encryption.) So I entered my passphrase and then it proceeded to decrypt the file and store the decrypted file into the file ccards.dec. That's all well and good. Did it work? Of course. I can visually verify, if nothing else, that the original file, ccards, and ccards.dec are the same. But I am not going to show those files to you here, of course. But I will show you the encrypted version of the file. I have faith in this program and in the theory enough that I know that you will not be able to crack my file and get my credit card numbers. Here is ccards.asc.

Basically, the information is thought to be secure to the extent that the processing resources required to break the code would be too much for any one individual, or even organization to muster. And even if they could muster tremendous resources at the task, it would still take a long time. The specifics are beyond me but I do have confidence.

I meant to go back and show you two more ways you might prefer to use the encryption command line:

```
[Wed May 31 12:44:22][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--armor --recipient glmull --sign --output ccards.asc --encrypt
../../ccards You need a passphrase to unlock the secret key for
user: "Glenn Mullikin (hello) <glmull@altavista.com>"
1024-bit DSA key, ID 91326DD5, created 2000-03-25 [Wed May 31
12:44:48][glenn_m localhost][jobs 1][~/1test/gpg]$
```

The above command specifies the recipient in the command line and so we don't need to give it on stdin, this allows for almost automated execution, with the exeption that we still need to enter the passphrase if we are digitally signing a document (with the --sign option).

The other way you might run the encryption is as follows:

```
[Wed May 31 12:47:10][glenn_m localhost][jobs 1][~/1test/gpg]$ gpg
--armor --recipient glmull  --output ccards.asc --encrypt
../../ccards
[Wed May 31 12:47:18][glenn_m localhost][jobs 1][~/1test/gpg]$
```

Without the --sign option, you get it done without any further user input. And this leads me to the end of today's look at GnuPG.

Another thing that can be done is encrypting binary files. I have encrypted jpegs. It works the same exact way, just replace the filename in the above examples with whatever image file you want to encrypt. Yes, you can produce an ascii encoded encrypted file of a jpeg, just by using the --armor option. Pretty useful if you want to email jpeg files back and forth or whatever image file I would think. Maybe you can perform some analysis on file size expansion when encryption is performed on such files.

Resources

It's a very easy download to get gnupg. Just go to the GnuPG website (http://www.gnupg.org/) and then you need to compile and install using the traditional tools. And then it works! You will first have to generate your own public/private key pair but this is very easy using the command line option --gen-key.

Counterpane.com (http://www.counterpane.com/) is a pretty good resource.

This article is re-printed with permission. The originals can be found at:

http://www.machineofthemonth.org/articles/a12/index.html

The Next Generation of Programming: Programming as an Engineering Discipline

Authors: Juris Reinfelds (New Mexico State University, Las Cruces, NM, USA) and Peter Van Roy (Universite catholique de Louvain, Louvain-la-Neuve, Belgium)

For developing lasting programming skills in the next generation of computer professionals, this panel discussed how programming can be taught as a true engineering discipline. This panel, consisting of Juris Reinfelds (moderator), Peter Van Roy, and Tiajaru Diverio, led an animated discussion of which a summary follows. The panel was recapped and expanded during a discussion session held subsequently.

An engineering discipline consists of a set of practical techniques and standard guidelines, firmly reposing on a scientific theory. For example, bridge building is based on Newtonian mechanics and materials science. Teaching an engineering discipline consists of two parts: teaching the fundamental concepts (the science) and teaching the current tools (the technology). Knowing the science prepares the student for future developments. Knowing the tools prepares the student for the present.

PROGRAMMING IS NOT TAUGHT IN THIS WAY

We de ne programming broadly as the step from speci cation to running program, which consists in designing the architecture and its abstractions and coding them into a programming language. Up to now, programming has been taught more as a craft than as an engineering discipline. Usually, programming is taught in the context of one (or a few) programming languages (e.g., Java, complemented with Haskell or Scheme). The historical accidents of the particular languages chosen are so closely interwoven with the fundamental concepts that the two cannot easily be separated. There is a confusion between tools and concepts.

Teaching programming in this fashion is like teaching how to build bridges built only of wood or iron. Engineers would implicitly consider the restriction to wood or iron as fundamental and would not think of using other materials or even of using wood and iron together. This restriction has led to di erent programming communities, based on di erent programming \paradigms": object-oriented, logic, functional, and so forth. The unity of programming as a single discipline has been lost.

The result is that programs often su er from poor design. We gave an example based on Java, but the problem exists in all programming languages to some degree. Concurrency in Java is complex and

expensive, so its use is discouraged. Java-taught programmers reach the conclusion that concurrency is always complex and expensive.

Program specifications are designed around the concurrency restrictions, often in a contorted way. But these restrictions are not fundamental at all. There are forms of concurrency that are quite useful and yet as easy to program with as sequential programs. Furthermore, it is possible to implement threads almost as cheaply as procedure calls. If the programmer were taught about concurrency in the correct way, then he or she would be able to write speci cations for and program in systems without concur- rency restrictions (including improved implementations of Java).

THE KERNEL LANGUAGE APPROACH

Current languages are the result of over ve decades of language design e orts. They are capable of scaling to programs of millions of lines of code. In our view, this success is not completely fortuitous. It is because they model some essential aspects of how to construct complex programs. In this sense, they are not just arbitrary constructions of the human mind. They merit scienti c study.

How can we separate the fundamental concepts in these languages from their his- torical accidents? The kernel language approach shows one way. The full language is translated into a small kernel language that has a minimal number of programmer- signi cant elements. This gives the programmer, i.e., the student, a clear insight into what the language does. The kernel language has a simple formal semantics, which gives a solid foundation to the programmer's intuition and the programming techniques built on top of it.

Reducing a complex phenomenon to its primitive elements is characteristic of the scienti c method. It is a successful approach that is used in all the exact sciences. For example, materials science explains the behavior of bridges in terms of simple concepts such as force and energy and their conservation laws.

Earlier attempts to make programming into an engineering discipline have failed. Let us investigate why. The two approaches that have been used to de ne languages precisely are the foundational calculus and the virtual machine. A foundational calculus, like the Lambda-calculus or Pi-calculus, reduces programming to a minimal number of elements, regardless of whether these are relevant for practical programming. This is important for the theory of computation, but practice shows that it is largely irrelevant to programmers. A virtual machine, which de nes a language in terms of an implementation on an idealized machine, is useful for implementors. Its concepts are close to hardware and again are largely irrelevant to programmers.

THE TEXTBOOK

Peter Van Roy and Seif Haridi have written a programming textbook based on the kernel language approach. The textbook organizes programming at three levels:

- Concepts: compositionality, concurrency, encapsulation, lexical scoping, higher- order, data ow, state, inheritance, etc.
- Techniques: how to write programs with these concepts.
- Computation models ("paradigms"): data entities, operations, and a kernel language.

This view of programming explains most often-used programming techniques in terms of simple kernel languages. The kernel languages are introduced in a progressive way, by adding concepts one by one. All have a complete and simple formal semantics. Programming paradigms emerge in a natural way when programming, as a kind of epiphenomenon, depending on which concepts one uses and which properties hold of the program. The unity of programming is regained. Restrictions on the programmer's expressive power have been removed.

The textbook is being teach-tested in Fall 2001 and Spring 2002, in three universi- ties (KTH in Stockholm, Sweden, UCL in Louvain-la-Neuve, Belgium, and NMSU in Las Cruces, NM, USA), in both second-year and fourth-year undergraduate computer science courses. The expected publishing date is 2002. The latest draft of the textbook is always available at:

http://www.info.ucl.ac.be/people/PVR/book.html

The textbook is complemented by a full-featured open-source software package, the Mozart Programming System, that can run all program fragments in the book. All information about Mozart including binaries for downloading is available at:

http://www.mozart-oz.org

This article was submitted to the recent AUUG conference, and is is printed with permission.

Nessus : another brick in the (security) wall

Author: Georges Tarbouriech <georges.t@linuxfocus.org>

Abstract:

Nessus is a free security scanner available from http://www.nessus.org. The project was started and is maintained by Renaud Deraison. The stable version at the time of this writing is 1.09 and the experimental one is 1.14. The software is released under GPL and many people contribute to the project, especially for plugins... while some other people benefit from nessus work without even mentioning the name (more on this at the end of the article). Nessus works on many Unix flavors as a client and a server, and on Win32 as a client. Let's have a look at this great tool.

Getting and installing nessus

Going to http://www.nessus.org, you can get this great piece of software. Since nessus is also available as a client for Win32, we obviously will consider the Posix version in this article.

To use nessus, you need at least nmap and Gtk (Gimp Toolkit). Links to those tools are provided from nessus website. However, since you can use nessus from the command line, Gtk is not mandatory.

You can get nessus in three different ways : the good, the bad and the ugly.

The good way is the standard one, that is, you download the archives from the ftp site closest you. You have four archives : nessus libraries, nasl libraries, nessus core and nessus plugins. Once unpacked, you build and install them as usual : ./configure, make, make install, following the above archives order. If you have a previous version of nessus installed on your machine, you'll have to remove it. To do this, nessus provides an uninstall script to use after the first "./configure" in the nessus libraries package. Run this script before typing "make". Do the same for each provided package (except running the uninstall script) and you're done.

The bad way, consists of running a downloadable script called nessus-installer.sh. Then typing "sh nessus-installer.sh" will auto-install the package. The four packages don't need to be installed separately. It's now just one standalone package.

The ugly way : as long as you have lynx installed on your machine and you're connected to the Internet, just type "lynx -source http://install.nessus.org | sh" and that's it. You must NOT be root to do this.

Obviously, we recommend the "good" way... well, if you download nessus, we suppose security matters to you ! Since we're talking about security, don't forget to check the MD5 checksum.

Nessus comes with different utilities (nasl, a scripting language, nessus-adduser, nessus-build...). Each of these utilities has its own man page for the client and the server. More documentation is available within the distribution (README, INSTALL...) or on the nessus website.

Configuring and running nessus

To make things easier to understand, we'll show examples with the nessus X11 version, the one that uses Gtk.

The nessusd server

To start nessus, you obviously need to run the server daemon, nessusd. When launching the daemon for the first time you'll have to create a username and password by using the nessus-adduser command. If the nessus libraries package has been compiled with the "--enable-cipher" option (highly recommended, not to say mandatory!), nessus generates a private key. This key can be protected with a passphrase. The server has many options available and you'll find them all in the nessusd man page.

From there you can create the user database and the corresponding rules. That determines who may to run the server daemon and what you will allow her to scan (a machine, a network...). The rules are of the form "accept" or "deny" followed by a network IP address with its netmask.

For example: accept 192.168.1.0/24, allows the user to test the whole 192.168.1 network.

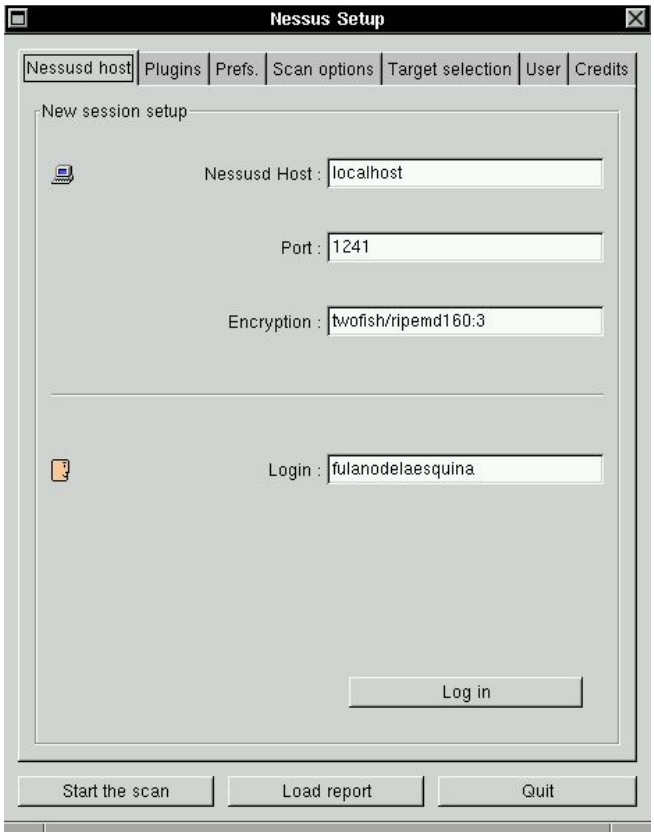
It's also possible to define one single user with no rules at all. If you wish to allow various users to run nessusd, you'll have to be very careful about what you allow them to do. You can't let everybody do everything on your network, can you ?

Last, nessusd relies on a configuration file (usually) found in /usr/local/etc/nessus/nessusd.conf. You can change this file by hand - as soon as you know what you're doing.

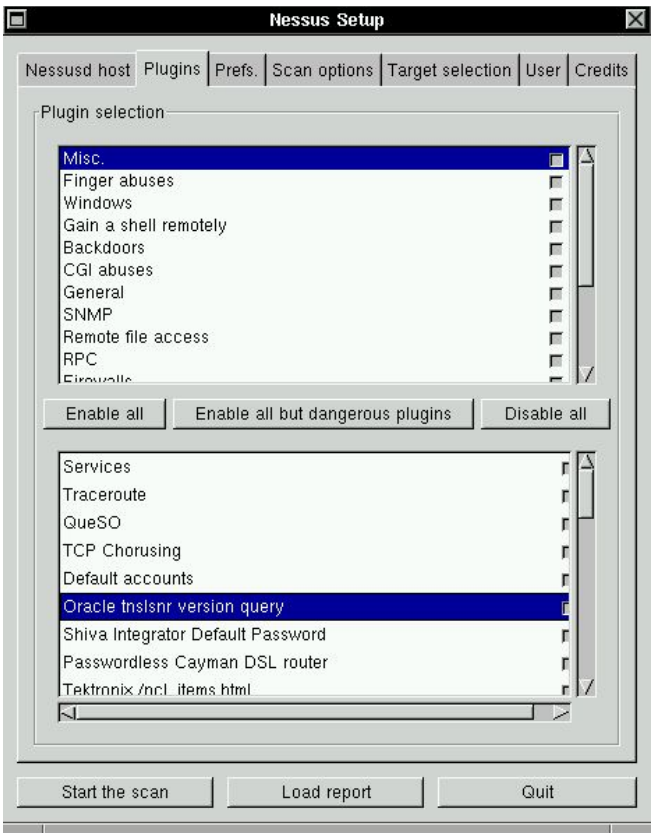
The nessus client

After configuring and starting the demon, you can start the nessus client to connect to the nessusd server. One way to run the client is to type "nessus &" in a shell. This opens the nessus setup window after asking for the above mentioned passphrase. This window provides you with seven tabs.

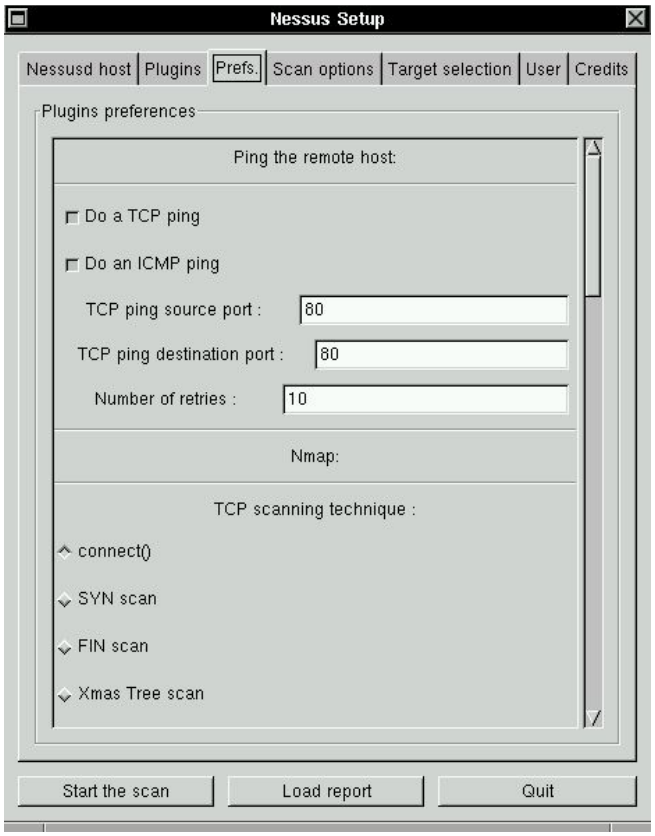
The first tab is called "nessusd host". From it you can connect to the nessusd host clicking on the "Log in" button. Of course, this assumes you're allowed to connect as this user, in other words, that your username exists in the user database.



The second tab concerns the plugins. Here you select or deselect the plugins you want to use during the scan. For instance, you can disable the dangerous plugins (the ones able to crash a machine!). Clicking on a plugin in the bottom part of the window displays some information about that plugin.

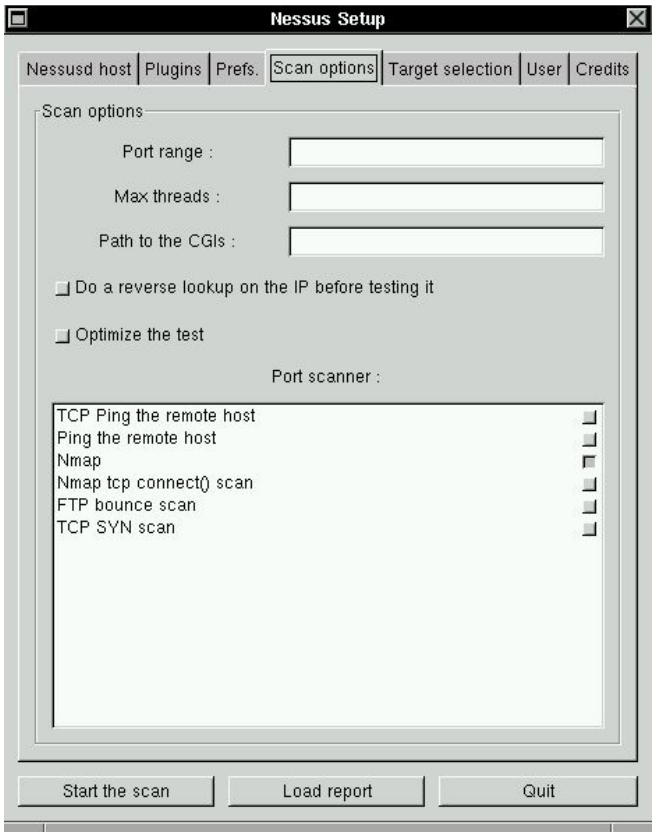


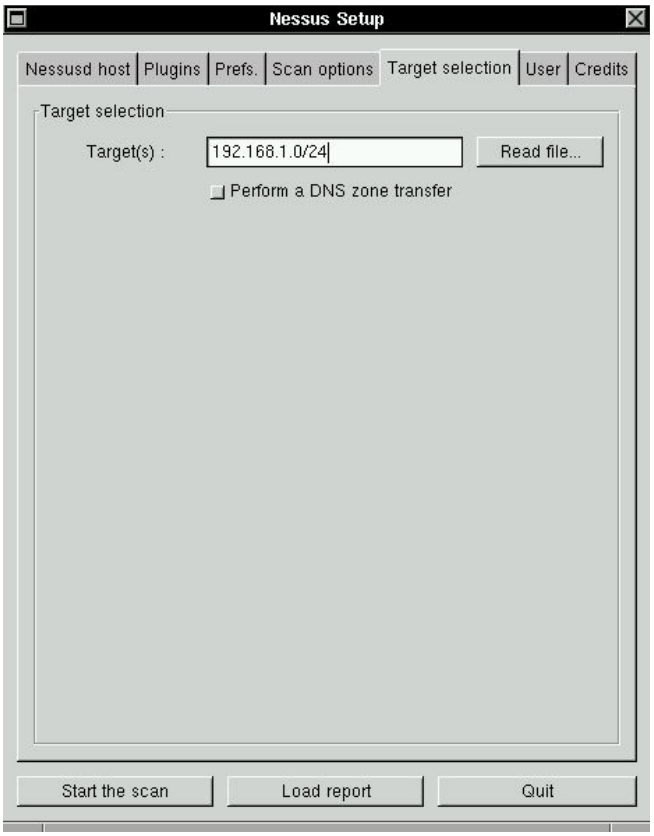
The third tab allows you to define the preferences for the plugins. This concerns ping, TCP, FTP... Here you can fine tune the way you'll use nessus to scan the target host(s) or network.



The fourth tab allows you to define the scan options and the port scanner to use, usually nmap. Find out more on nmap there: (http://mercury.chem.pitt.edu/~tiho/LinuxFocus/English/July2001/article170.shtml).

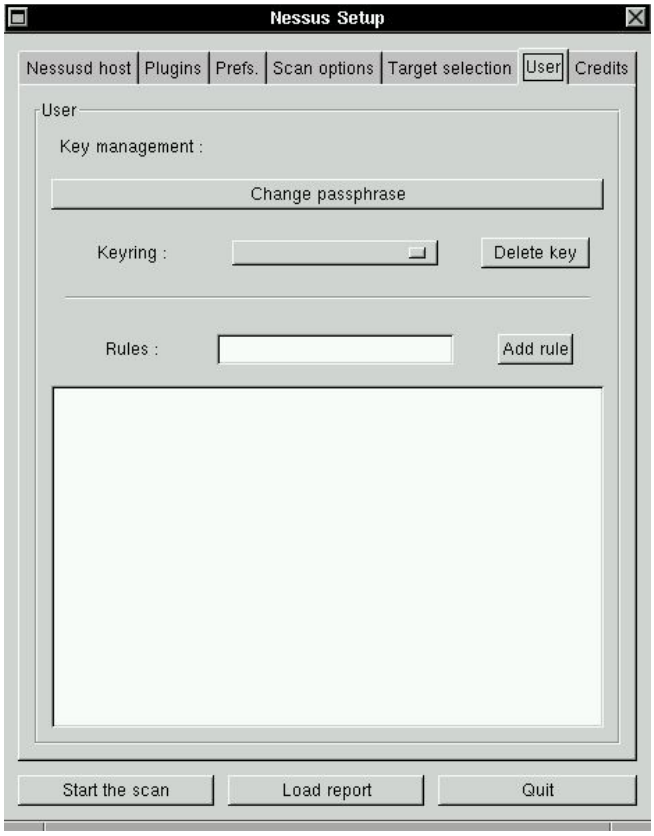
[Editor's note: NMAP is also covered later in this issue]



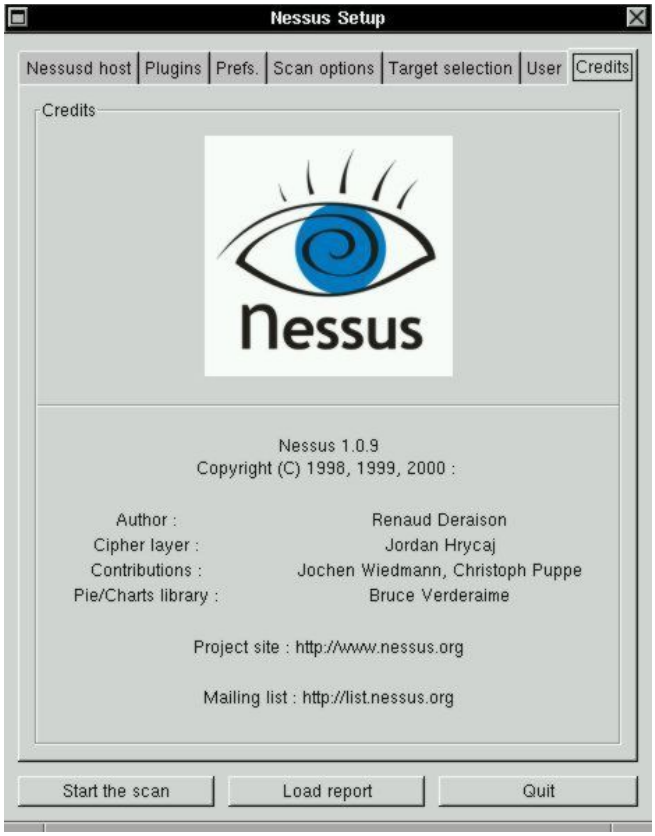


The fifth tab is where you tell nessus the target of your scan. In the target field you can write the name of a host, the name of different hosts separated by commas, one or more IP addresses, again separated by commas, or a network address with its netmask (for example 192.168.1.0/24). There's also a check box to perform a DNS zone transfer. That is, connecting to a DNS server, nessus will try to get the list of the hosts in this domain.

The sixth tab allows the user to change his passphrase, to delete his private key or to add rules.



Last but not least, the seventh tab opens the credits window containing as well the version number. That shows all the information you should provide when using nessus for a different project... Well, that's the way it should be!

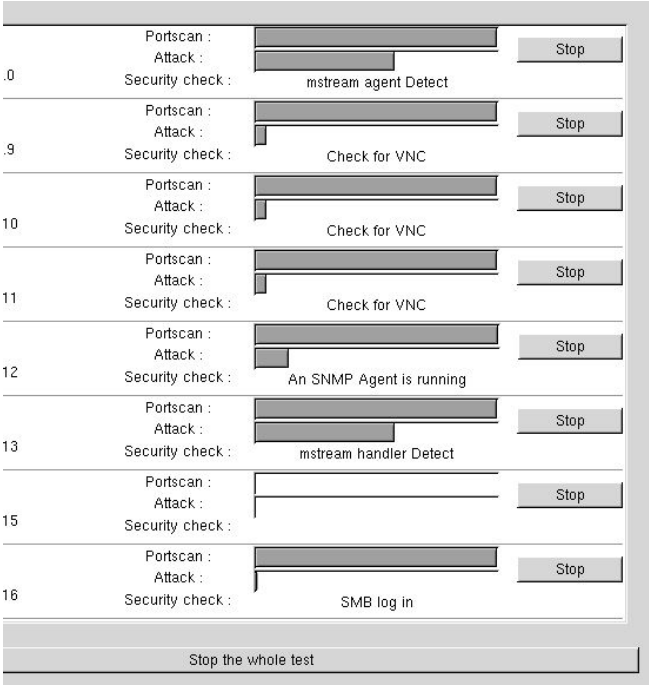


You can have an eighth tab if you compiled nessus with the "--enable-save-kb" configure option. Kb stands for "knowledge base". This experimental feature allows to using the results from previous tests. This feature will be a default one in nessus 1.1.0. More on kb at
www.nessus.org/doc/kb_saving.html

Once you have "visited" every tab, you can run the "beast". Click the "Start the scan" button. What happens then ?

NESSUS AT WORK

When you start the scan, nessus opens a window displaying the scan status. For example, let's say you are testing a whole network, called 192.168.1.0/24. Eight machines (hosts) will be displayed at once, showing which plugin is used for which machine and a progress gauge. It looks like this:



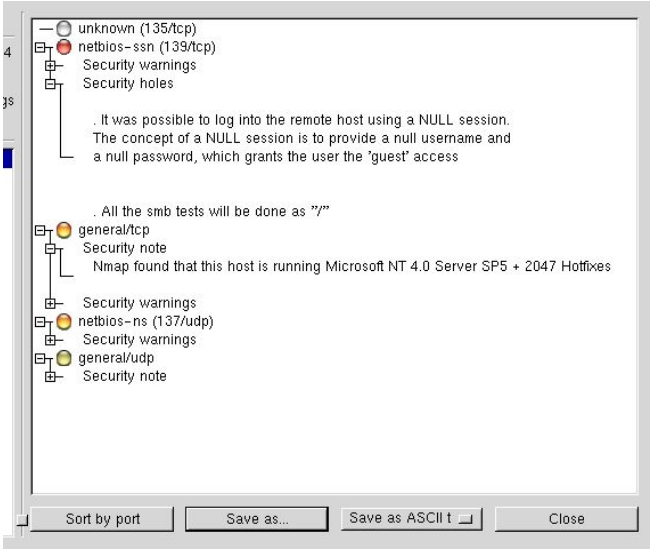
As you can see, the whole test can be stopped at any time.

Obviously, if you scan a whole network with a lot of hosts, the test will last quite a long time. It will depend on the OSes, the network speed, the machine's roles (more or less open ports), the number of active plugins, etc.

You can also test in two other different ways : the detached scans or the differential scans. This assumes you compiled nessus with the above mentioned "--enable-save-kb" configure option. The detached scans allow running the tests in the background while the differential scans, as the name says, only shows the differences between two scans.

You'll find much more information about these features going to nessus documentation (www.nessus.org/doc).

Of course, the result you get at the end of the scan is the most important. One of nessus' greatest features is reports it provides you with.



These rather detailed reports often suggest a solution for the detected vulnerability. Even more, they really reliable. If a found vulnerability is not a real one, nessus tells you that it might be a false positive. This can happen, for instance, with patched versions of some daemons : a recently corrected vulnerability may be detected as a potential risk. However, the plugins are quickly updated for this sort of thing.

Another small mistake may come from nmap (2.53) when identifying the OS version. But, this is really of little interest. Personally, I don't mind if NotTerminated 4.0 with SP6a is identified as NotTerminated 4.0 with SP5, or if Linux kernel 2.2.19 is detected as 2.2.14. I won't either complain about "exotic" OSes such as AmigaOS or BeOS identified as a printer or a router. I mean, I can't imagine sending a mail to Fyodor (nmap's author) to tell him such a thing : who uses such OSes in a network today ? For the AmigaOS, I would say 5 people... in the whole world :-(

Some other OSes are not perfectly identified either, but they often are rather "new" or not really used like MacOS X or QNX. But, again, it isn't that important and this may be already solved in the new 2.54 beta version of nmap (and, by the way, this new version provides a MacOS X port).

Anyway, the main point is that nessus gives you tons of information that allows you to correct the machine's vulnerabilities or weaknesses on your local network. These reports can be saved as text, NSR, HTML, HTML with pies... thus allowing comparisons between two scans. It may seem obvious, but the state of a network at a given time may be quite different from what it can be 30 minutes later. Why ? A network is alive! This is one of the main reasons why securing a network is not easy : things change all the time. If you're wondering about the need to use tools such as nessus and nmap, that is the answer. If you are curious about the way nessus works, watch the system logs or, if you use snort, watch snort logs. Another place to find information is in /usr/local/var/nessus.

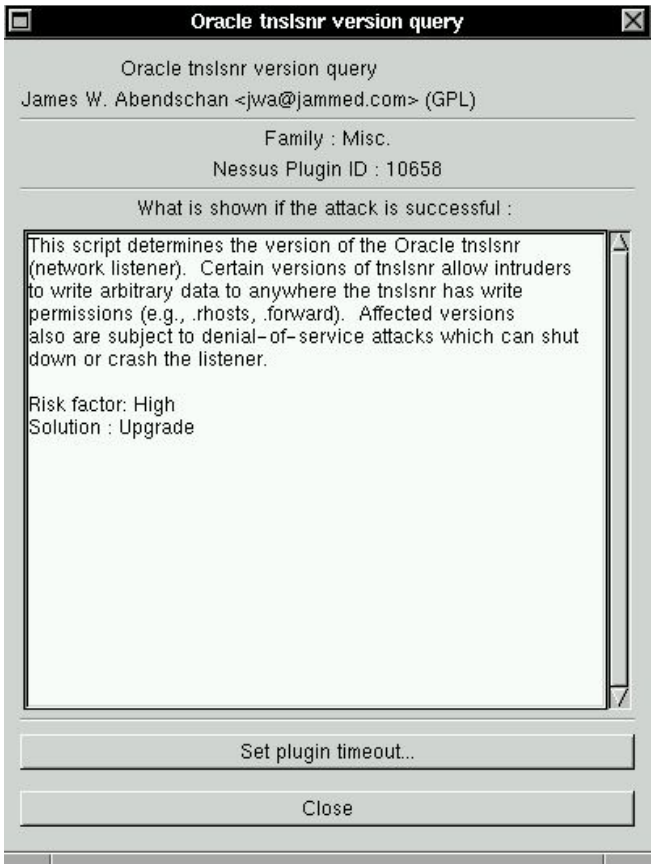
From there, you probably will have some work to do to reduce the weaknesses of many machines in your network. The more you harden each host, the better. To help you in this big task, nessus (and nmap) are tools you can't live without.

Plugins

Plugins are the "heart" of nessus. They are security tests - that is test programs to discover a given vulnerability. NASL (Nessus Attack Scripting Language) is the recommended language to write security tests. You'll find a lot about NASL going to this URL : http://www.nessus.org/doc/nasl.html.

Accordingly, if you want to contribute to the nessus project writing plugins, this is where you'll find the right information. At the time of this writing, there are 756 plugins in nessus' database!

There are almost 20 plugin families: backdoors, denial of service, gain root remotely... As already mentioned, each plugin reports information. It tells you what's wrong and what you should do to correct the problem.



We can't talk about plugins without mentioning CVE (Common Vulnerabilities and Exposures). It's a huge information database available from http://cve.mitre.org There you'll find all the details about known security risks. Another great place to share knowledge. This website is the absolute

reference that you must visit.
Of course, there's a lot to say about nessus plugins, but a book wouldn't be enough. A good way to understand how they work and how they are written is to read them from your /usr/local/lib/nessus/plugins directory.

Thanks again to Renaud Deraison and contributors for doing such a great job.

AND NOW FOR SOMETHING COMPLETELY DIFFERENT ...

Even if this title sounds like Monty Python, unfortunately there is no humor in it. The three or four people reading my articles know about my usual off-topic sentences : this time, it's a whole section ! Is it really off-topic, that's another story. Let's go.

Since I'm quite interested in computer security, I often visit the dedicated websites. Sometimes to learn about new vulnerabilities or to discover new security tools. Incidentally, I found a few products based on an online scanning service. In fact they call themselves (at least for now) ASP (Application Service Provider). If you go a bit further, you quickly discover that the engine behind the service is nessus. So far, so good. However, when trying to find the explicit information you can't see the word "nessus". That's where I'm hurt!

Many people working for the free software community do this work for free, not for a living. The only payment they get is called : credit. Nessus is released under the GPL. That is, everyone can use the product, modify the source code, adapt it... as long as they mention the original author(s). Of course, the license says much more than this. If you don't know the GPL, have a look there.

To me, this looks like a theft. I mean, I don't even need a license to credit other people's work. I have very much respect for those people working for free (almost always) and sharing with a community. They deserve recognition from that community. This is especially true when people try to make money from somebody else's work. You can call that respect, recognition, it doesn't matter. The fact is, those words seem to have lost their meaning. Sure, we could say the GPL is the cause of such behavior. With ASP, you're not considered to be selling the software. Depending on the country, the GPL may have no legal value.

A solution could be that the users of such ASPs request the name of the scanning engine doing the real work. The answer given will at least show if the people providing the service are "honest". If they don't answer or if they say they "invented" that engine (which you identified as nessus), just don't use it !

Install nessus instead (the true one), it'll be more secure anyway. But again, do we need a license to say "thank you" to people having done a really big and great job ? And, by the way, Renaud Deraison doesn't want to change nessus license : nessus will stay under GPL.

Sorry for that long digression, but I believe it had to be said. It's all over!

Despite the last section, what to remember from this article is that nessus provides a high quality standard. It's quite an impressive software. Used in conjunction with nmap, it becomes a must have whenever security is a concern. It is a very responsible tool, improving every day. Thanks to Renaud and friends for their constant updates to plugins.

Today, a sysadmin can't work without nessus and nmap. These tools find vulnerabilities you thought you solved. This is true for most of the OSes found on your network. And when you know that some OSes are like sieves, nessus lets you relax a bit.

Even more, nessus can help you understand the way a network (or a machine) can be compromised. If you read the reports provided carefully, take them into account and make the right corrections, you'll improve the security of that network (or machine) considerably. Again, I said "improve" : your network won't be 100% secure just because you run nessus. The road to security is a very long one and we are far from the end of it. Once more, thanks to the free software community for the great work it does about security.

Concerning those nice people trying to make money from the work of this free software community members, I'd like to add something. Saying "thanks" is not a badge of shame. Being honest is not that awful, is it ? If this kind of behavior grows, the risk is either the end of the community or a big change in licensing (and probably more and more patents !). In either case, you'll be on your own and things will become much harder for you. And unfortunately, we won't be able to use free software anymore. This doesn't mean you'll be able to sell yours. Think it over!

Aren't we living in a great time?

http://mercury.chem.pitt.edu/~tiho/LinuxFocus/English/November2001/article217.shtml

---

SUS – An Object Reference Model for Distributing UNIX Super User Privileges

Author: Peter Gray, Information Technology Services, University of Wollongong, <pdg@uow.edu.au>

INTRODUCTION

It has long been known that the "all or nothing" traditional UNIX security model where the super user account (usually "root") has practically unlimited administrative powers while normal users are subject to all security restrictions can be a serious inconvenience for large sites with multiple system administration staff.

Delegating areas of responsibility to junior system administration staff or allowing certain users to perform limited administration tasks is very difficult. This often results in large numbers of staff knowing the "root" password or the installation of a large number of "setuid" executables to allow staff to perform tasks requiring "root"privileges. Both of the above result in reduced levels of system security and higher than desired system administration overheads.

There have been many tools written to allow restricted access to commands which may be run as "root", SUDO, PRIV and SUPER to name a few. These tools let a system administrator allow a user to run a single command as the super user, usually after performing a secondary authentication step such as supplying their normal UNIX password.

These tools consult a configuration file to determine if the invoking user is allowed to run the command being requested (the target command). The determination is usually based on simple pattern matching of the command and arguments. While this method works well in many cases there are some situations where it does not provide the desired degree of fine-grained control over what a user may or may not do as "root".

SUS is a tool to permit a system administrator to delegate operations that a user may perform as "root" based not on what the target command looks like but on what objects the command will operate on.

The configuration file defines for each user which target commands they may execute as root (or as any other user). Additionally, the configuration file controls many aspects of the operation of SUS.

SUS treats the command itself and optionally any arguments which are present as references to system objects. Objects can be such things as hosts, files, processes, users or groups. Such system objects have attributes, such as in the case of files, owners, groups, size etc.

The actual attributes which are present vary with differing object types. For example, files have owners, users have home directories.

Referenced objects themselves may contain references to other objects. For example, a process object has an attribute which is the effective user id. The effective user id is a reference to a user object. The user object thus referenced will have a home directory as one of its attributes. That home directory is a reference to a file object and so on. The configuration file syntax allows for such recursion.

CONFIGURATION FILE

As the configuration file is read it is preprocessed line by line via a CPP style preprocessor. This allows the system administrator to define macros to simplify the configuration file appearance. The addition of some predefined macros along with the conditional capabilities of CPP allows the overall syntax of the configuration language to be simplified. For example, there is no need for the configuration language to allow for restricting commands to certain times of day since this can be easily achieved by using the CPP conditional directives in combination with the predefined macros containing the current time.

In addition, certain macro names are used to control the internal operation of SUS. For example, the location of the SUS log file can be set by defining the macro "LOG_FILE". Nearly all aspects of the internal operation of SUS can be controlled in this way. This removes the necessity for a large number of compile time configuration parameters.

When a user requests SUS to run a command, SUS will read the configuration file looking for entries which apply to the current user. As such entries are found, the actual command, including arguments if present, is matched against the commands allowed by the current entry being processed. Arguments are matched only if the current command from the current entry specifies argument matching. It is possible to specify that the command must have no arguments or that certain arguments must be matched but the rest can be ignored or that all arguments must be matched.

The entry may define allowed commands and arguments as simple regular expressions or as references to objects. If the latter, the entry will contain a list of selectors which are used to determine if the actual referenced object(s) belong to the permitted set. If they do, the command is allowed, if not SUS continues to process additional entries in the configuration file.

If at any point as the command is being tested for validity a match is found with any selector but the logical NOT operator was prepended to the selector, processing is terminated and the command is not allowed.

Appendix A describes the configuration file syntax.

Object Classes

SUS contains the built-in ability to match certain types of system objects, including users (as defined by entries returned by getpwent(3)), groups (as defined by entries returned by getgrent(3)), files (as defined by the result of a stat(2) system call), hosts (as defined by the values returned from gethostbyX(3) and getipnodebyX(3)) and processes (as defined by the psinfo entry in /proc).

Each of the above object classes takes a list of selectors which can be used to match on the attributes of the referenced object. In cases where an object contains a reference to a different object (such as processes effective user id referencing a user), a selector exists to perform matching using the referenced object.

The full list of selectors available for each of the object classes is listed in appendix B.

Appendix C is a annotated sample SUS configuration file which shows some of the capabilities of the tool.

Environment Control

SUS allows the environment of the target command to be run to be completely controlled by the configuration file. The environment may be completely replaced, selected entries removed or added or left completely unaltered from that of the invoking user. In addition, the control of the environment can be different for different users or target commands.

Logging

SUS will normally log each invocation, regardless whether the target command was allowed or not. Logging can be either by syslog(1) or direct writes to a file or both. Information logged includes the invoking user, the current directory and the target command with arguments. Optionally, SUS can wait for the target command to complete and log usage information including start and end clock times and system accounting information.

Timestamps and authentication

Upon each successful invocation, SUS records a timestamp in a system directory (by default the root directory) and the user's home directory. When invoked, the timestamp from the system directory is retrieved. If it is valid it is checked for currency. If current, the user does not have to perform any authentication step.

If the system timestamp is valid but not current, the timestamp from the user's home directory is retrieved. If valid it is checked for currency. If current, no authentication step is required.

In all other cases except promiscuous mode (see below), the user must authenticate using their own password. In promiscuous mode, the password of the target user is used for authentication.

A SHA1 checksum is used to prevent tampering with timestamps stored in the user's home directory. The checksum is computed using information stored in the system timestamp which is then removed from the home directory timestamp before it is written. All timestamps become invalid if the user changes their password.

Promiscuous mode

Promiscuous mode allows the invoking user to run a command as another user (not root) if the invoking user can suppy the target user's username and password. Which users who are able to use promiscuous mode and which commands they can run is controlled by the configuration file.

This mode is designed to ease the provision of services to users via the WWW. In many cases you have a command on the system (often setuid) to perform a service for the users. An example is the passwd(1) program to change a user's password. The password program can be used by a user to change their own password but nobody elses. With the web server running as a non-privileged user (for example user "APACHE"), it is difficult to provide a password change service without adding another setuid binary.

By using promiscuous mode, the web server user can use SUS to run a command (such as passwd(1)) as another user. The user can supply their own user name and password which SUS will use to determine that user "APACHE" can execute /usr/bin/passwd as any user if the username and password of the target user can be supplied. Thus a user can run the password command as themselves via the web after authenticating without the installation of additional setuid binaries. This reduces the number of setuid binaries which must be installed and maintained.

Miscellaneous Features

SUS also supports the following functionality:

- The ability to run the target command at a specified nice(2) value.
- The ability to run the target command in the background as a session leader.
- An option to change to a specified directory before running the target command.
- The ability to run the target command as any user (specified in the configuration file).
- A method to determine if a specified user is allowed to run a specified target command (super user only).
- A option to suppress initialization of supplementary groups for the target user.
- A shorthand option for starting a shell.
- An option to not update timestamps and to invalidate timestamps.

Conclusion

SUS extends the ability to delegate super user privileges by treating command arguments as references to system objects and allowing the requested command to be allowed or rejected based on the attributes of the objects the command will operate on. This allows a finer grain of super user access delegation to be implemented than systems using simple matching of command names and arguments as strings.

Availability

SUS is currently in alpha testing. A beta version is expected to be released in October 2001. When available, it will be found at http://pdg.uow.edu.au/sus

SUS is being developed under solaris 2.8 but will be ported to linux before release.

Appendix A – Configuration File Syntax

After preprocessing, the configuration file contains lines with the the following syntax: Each directive is processed one at a time, in the order they appear.

```
directive ::= user_list : command_list

user_list ::= full_user_expression {, ...}

command_list ::= full_command_expression {, ...}

full_user_expression ::= user_expression{@host_expression}

user_expression ::= username_re ::= user_class ::= rexexp_class

username_re ::= {!}extended_regular_expression
host_expression          ::= host_re
                         ::= host_class
                         ::= regexp_class

host_re ::= {!}extended_regular_expression

full_command_expression ::= {~target_user_expression:}command_expression

target_user-expression :== user_expression

command_expression       ::= command_name_re {argument_expression}...

command_name_re          ::= {!}extended_regular_expression

argument_expression      ::= argument_re
                         ::= class

argument_re ::= {!}extended_regular-expression

class                    ::= file_class
                         ::= group_class
                         ::= user_class
                         ::= proc_class
                         ::= host_class
                         ::= regexp_class

user_class               ::= USER(selector_list)
                         ::= USERNAME(selector_list)
                         ::= USERID(selector_list)

file_class               ::= FILE(selector_list)
                         ::= RFILE(selector_list)

group_class              ::= GROUP(selector_list)
                         ::= GROUPID(selector_list)
                         ::= GROUPNAME(selector_list)

proc_class               ::= PROC(selector_list)
                         ::= RPROC(selector_list)

host_class               ::= HOST(selector_list)

regexp_class             ::= REGEXP(selector_list)

selector_list            ::= selector_argument{,selector_argument}

selector_argument        ::= {!}attribute_name = attribute
attribute_name           ::= string
attribute                ::= string
```

Selectors by class

USER Class

The USER class supports the following attribute selectors. The reference can be either a username or a userid.

Attribute	Name Matches
exists	"true" if user exists, "false" otherwise
uid	user id of the
user	name user name of the user
gid	primary group id of the user (string)
pgroup	primary group of the user (class)
ingroup	all users groups (class)
gecos	gecos field for user
dir	home directory of user (string)
home	home directory of user (class)
shell	shell of user

The class USERNAME works the same as class USER, but the reference must be a username. The class USERID work the same as class USER but the reference must be a userid.

GROUP Class

The GROUP class supports the following attribute selectors. The reference may be either a group name or a group id.

Attribute	Name Matches
exists	"true" if group exists, "false" otherwise
gid	group id of group
groupname	group name of group

The class GROUPNAME is identical to class GROUP, but the reference must be a group name. The class GROUPID is identical to class GROUP, but the reference must be a group id.

FILE Class

The FILE class supports the following attribute selectors. The reference must be an absolute or relative path name.

Attribute Name Matches

exists	"true" if file exists, "false" otherwise
name	name of file as supplied
realname	real name of file if available
type	type of the file (see below)
uid	user id of the file
owner	account of file owner (class)
owner	account of owner (class)
gid	group id of the file
group	group of file (class)
dev	device on which file resides
rdev	raw device on which file resides

File types are: reg (regular files), dir (directories), chr (character special), blk (block special), door, link (symbolic links) and fifo (named pipe).

The class RFILE works in the same way as class FILE, but if a match is not found with the file an attempt is made to match with with the file's parent directory and so on up the file system tree to the root. The search terminates as soon as a match is found.

The class PFILE work in the same way as class FILE, but all matching is performed on the parent directory of the file object referenced rather than the file object itself.

PROC Class

The PROC class supports the following attribute selectors. The reference must be a process id.

Attribute	Name Matches
exists	"true" if process exists, "false" otherwise
pid	process id of the process
ppid	parent process id of the process
uid	user id of the process
puid	user id of parent process
euid	effective user id of process
gid	group id of process
pgid	group id of parent process
egid	effective group id of process
sid	session id of process
argc	number of arguments of process
tty	tty device attached to process
owner	account of owner of process (class)
eowner	account of effective owner (class)
group	group of process (class)
egroup	effective group of process (class)

The class RPROC supports all the same attributes as class PROC, but if a match is not found with the file an attempt is made to match with with the process's parent and so on up the process tree to the root. The search terminates as soon as a match is found.

HOST Class

The HOST class supports the following attribute selectors. The reference can be a host name or an IP address.

Attribute	Name Matches
exists	"true" if host exists, "false" otherwise
name	matches name(s) of the host
v4_ip	matches IP addresses
v6_ip	matches version 6 IP addresses

Note that version 4 IP addresses are in dotted quad format (A.B.C.D) or classless network format (A.B.C.D/netmask_length).

REXEXP Class

The REXEXP class matches using extended regular expressions. It is present just for completeness and adds no new functionality.

Attribute	Name Matches
regexp	matches extended regular expressions

APPENDIX C – SAMPLE CONFIGURATION FILE

```
// Define the location of the SUS log file

#define LOG_FILE "/local/log/sus.log"

// Wait for child termination and log usage information

#define WAIT_FOR_CHILD 1

// How long to timestamp remain valid

#define MAXTIME 300 // 5 mins

// Delete these variables from the environment

#define ENV_DELETE "LD_.*"

// Change the SHELL variable

#define ENV_ADD 'SHELL=/bin/false"

// Set the path

#define PATH "/usr/bin:/usr/sbin"

// A class of people

#define SALES_ADMINS "jack | jill"

// A class of machines

#define SALES_MACHINES "proteus | hal"

// The sales people get control of their own machines
// but only during office hours
// ANY_COMMAND is a predefined macro as is HOUR

#if HOUR >= 9 && HOUR <= 4

SALES_ADMINS SALES_MACHINES : ANY_COMMAND #endif

// Define a class of users by virtue of their home directory

#define ENG_USERS USER(home=/home/eng/.*)
// These users can chown files to themselves in the engineering web
area
// SUS_USERNAME is predefined to be name of current user
// The RFILE selector matches recursively up the directory tree.

ENG_USERS : chown USER(name=SUS_USERNAME) FILE(exists=true, \
realname=/web/engineering/.*)
// As above, but add the restriction that the current owner
// of the file must be an engineer as well.

ENG_USERS : chown USER(name=SUS_USERNAME) FILE(exists=true, \
realname=/web/engineering/.*, \ owner=ENG_USERS)
// Allow Bill to run all the commands in a special
// directory we maintain for him with any arguments he likes.
// PFILE class works like FILE, but uses the parent directory
// for matching
// ANY_ARGUMENTS is a predefined macro

bill : PFILE(realname=/share/special/bill)

// Just to show NOT, allow everything for mary, except shells

SHELLS = .*/sh | .*/ksh | .*/bash | .*/csh | .*/zsh

mary : !SHELLS, ANY_COMMAND
```

This article was submitted to the recent AUUG conference, and is is printed with permission.

The Evolution of Debian Package Management Systems

Glenn Mullikin <glmull@machineofthemonth.org>

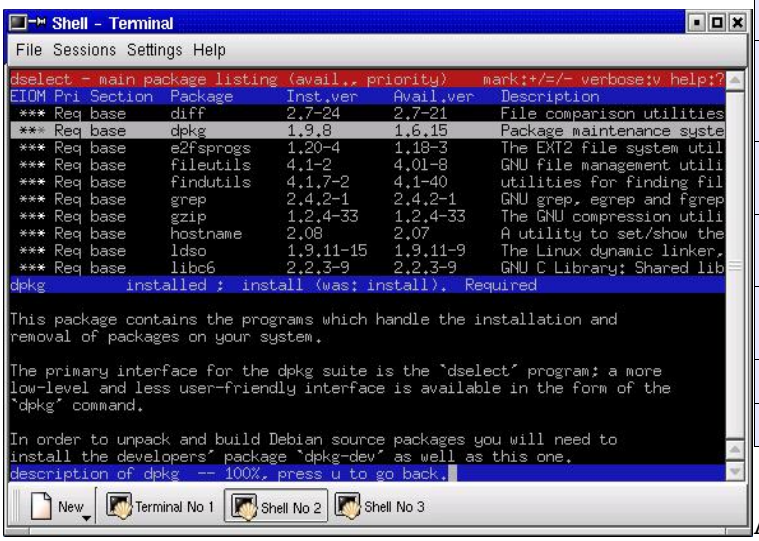
Introduction

It used to be that things were simple but not anymore. That program you want to install may require a ton of dependancies to be met. You can try and do that by hand but I'll bet you would like to have a way that it can be done for you, automatically. Got an internet connection? I want to tell you about some programs that can help you keep your system up to date.

I want to introduce you to some of the package management programs that I would like you to consider as a new user of Debian GNU/Linux. I have graphical user interfaces and I have some that run off a console. It's your choice and I am not going to try and make it for you.

So without further ado, let's get started shall we?

Dpackage (and Dselect)



Dpkg and dselect come in the same box (in the same package). So you get one with the other. I remember when I installed Debian for the first time and was shown the dselect tool and I thought it was so great because it showed a list of all the packages in some type of structural order so that I could see what was there for installing if I so desired. Typically, back then, I would use my cdrom drive as the access method for getting packages.

So Dselect originally was not meant as a tool for updating a system over the internet. Since then it has been expanded in functionality to use more advanced tools to do that job. However, for the new user that just came home with that new debian cdrom disc and wants to install it, dselect is probably the tool they will be using. Dselect will present them with a powerful and friendly user interface that allows them to select the packages they want on their system. It will let them remove packages and add packages.

Underneath dselect lies the dpkg management program which everything depends upon. The dpkg program can be run from a command line and you can learn about how to install and remove packages on an individual basis by typing "man dpkg" at the command prompt. And there may be times when you will need to go back to using dpkg to fix certain situations. I still use Dselect to this very day even though replacements are on the horizon. Dselect will probably always have a place in Debian to the extent that it supports non-internet based installs while many of the tools we will be discussing in this article are more geared towards pulling down updates off the internet only. There is nothing wrong with that because that is where everything is headed but Dselect lets you perform many different types of access methods. Take a look at all the different types of access methods it supports.

Different ways to install packages in Dselect

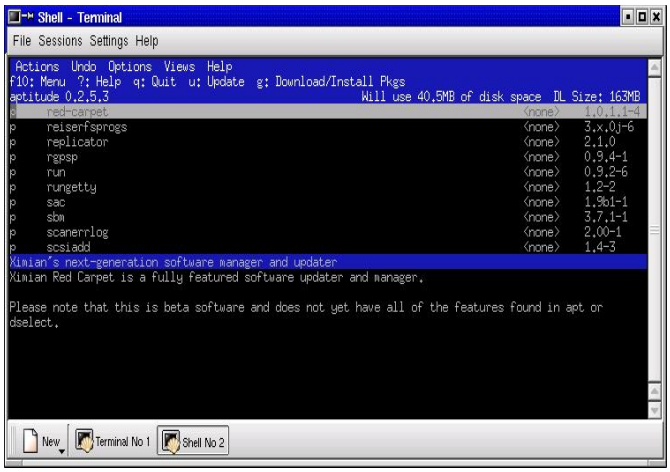
cdrom	Install from a CD-ROM.
* multi_cd	Install from a CD-ROM set.
nfs	Install from an NFS server (not yet mounted).
multi_nfs	Install from an NFS server (using the CD-ROM set) (not yet mounted).
harddisk	Install from a hard disk partition (not yet mounted).
mounted	Install from a filesystem which is already mounted.
multi_mount	Install from a mounted partition with changing contents.
floppy	Install from a pile of floppy disks.
apt	APT Acquisition [file,http,ftp]

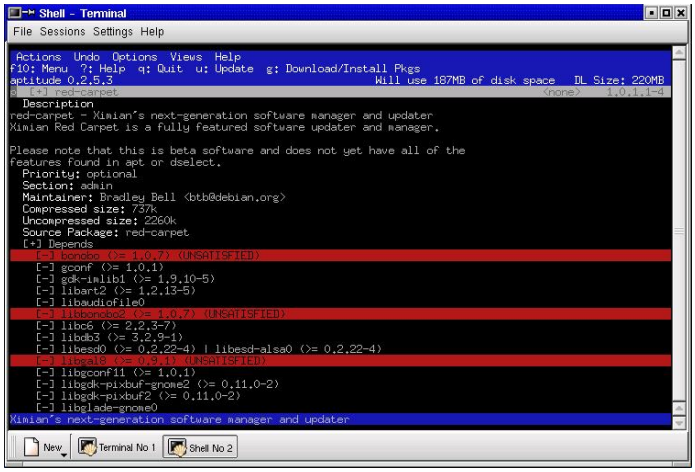
Aptitude

Apt is a tremendous package management system/tool. I have been using it for year or two and I love it. But we aren't talking about Apt here. We are talking about Aptitude. Of course, Aptitude is based upon Apt. Aptitude is a console/ncurses-based program which means that you don't need to have Gnome or KDE or any window manager installed. All you need is a command line. And maybe a few dependancies such as ncurses.

I installed Ximian Red Carpet using aptitude but it was not a very smooth thing. Unfortunately, I ended up upgrading a bunch of packages that I really don't think were really required to just install Red Carpet. I

think this may be because you have to use the "I" key to just select one package and any of its dependancies, while keeping all the other packages "on hold", as the user manual says.





Speaking of the user's manual, there is a very extensive one and it's very nice, although a little too technical for the newbie in many places, but certainly more advanced users wanting to get more out of the program will appreciate its awesome configurability (and the fact that someone took the time to explain the details).

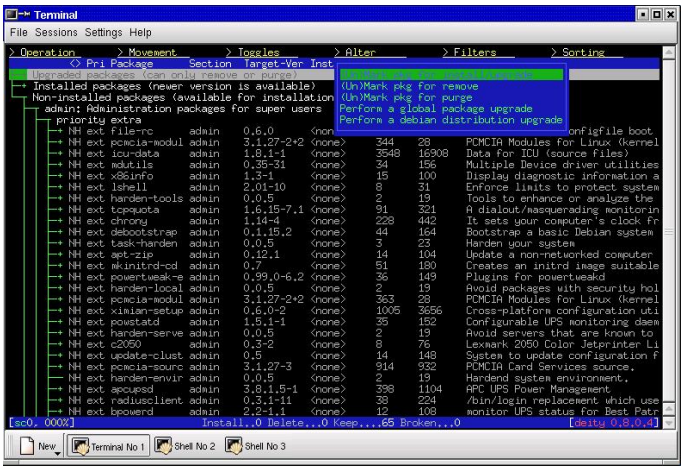
That's what makes Aptitude tick, all the powerful ways that you can have the program present things to you by the use of filters and sorting. I couldn't begin to explain it all but suffice it to say that the information is there to become a power user of Aptitude. As you can see in the above pictures, the program has the ability to present the user with a ton of details on every single package. When you hit the return button on a package it displays a new window with all the info on that particular package, a description, a list of packages that it depends upon and some other things. It even color highlights packages in red that need to be installed so you can see how many packages you will need to have installed before you can actually install this one. But don't worry, Aptitude uses Apt and so it has it all figured out as far as dependancies go. You just tell it what you want and it gets all the stuff that's needed.

I tried using the "I" key to install gabber and it didn't try and upgrade my entire system. It just installed the few dependancies necessary and then installed gabber too. So I like it now that I have figured out. The question I have is who is going to be annointed by Debian as being the official tool to use? Aptitude may not be the one. I have a feeling it may be the deity program. However, the feel I get from Aptitude is that it has maybe been around a little while longer and someone really put their heart into it so I'll continue to use it.

Of course, the help file is easily accessible by pressing the F10 key on your keyboard and tabbing over to the Help menu and selecting "Users Manual".

Deity-curses (Console-apt)

Deity is some powerful stuff. The reason I say that is that it has apt as its back end. But also, its ncurses user interface is very easy to learn and use. The program works very nicely in my experience. You can really appreciate the delicate attention to the user interface and how it presents the packages on the system that are installed and available for install to the user. You can do everything you would need to do directly from the user interface, no need to go to a command line to update the packages list. No need to go to the command line to do anything except type deity-curses as root.



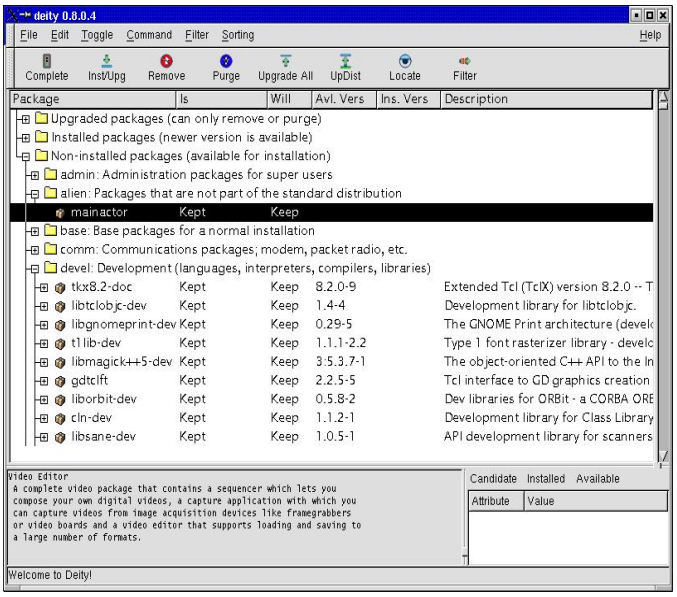
You can access all the needed functions through the main menu by hitting the capital letter associated with each name in the main menu. And then you can simply hit the left or right arrow key to move to different menus on the toolbar. Of particular note would be the Operation menu and the Sorting menus. The Operation menu lets you perform such things as Completing any pending installs or removals, updating the list of packages that are available, and a few other more obscure things. The Sorting menu is a great thing, once you figure out how it works. And once you do figure out how it works, then you can really do some really powerful package presentations on the screen that show you alot of good stuff. So this program is the bomb. It really is!

It's so easy also to get information on the package you are looking at. All you do is hit the "d" key and the screen splits into two and on the bottom half, you see a description of the particular package, a detailed description. Use the "[" and "]" keys to scroll that bottom window, if necessary. Want to know what packages are listed as dependancies for this particular package? Deity knows the answer and it will tell you if you will hit the Enter/Return key on your keyboard. Why does this work? Well, it's because of Debian's steadfast commitment to detail and hard discipline in the way that they do dependencies for their packages. It all starts at the bottom and works its way up. Start with good stuff and you get good stuff out. Start with bad stuff and you can't get good stuff out.

So Deity is just the result of alot of peoples' hard work and effort to make something good. Deity is not the one that does all that though. It is people. People that cared.

Deity-gtk

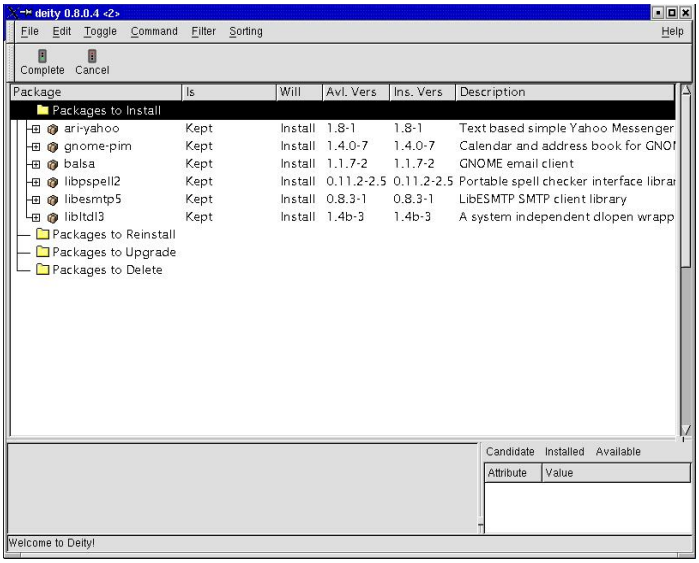
Deity-gtk is linked as a symlink to the deity executable but I guess the deity executable can figure out that it is being called to use the gtk graphical user interface, rather than ncurses. That being the case, here is a typical screen that shows deity-gtk in action.



I like Deity-gtk as far as the way it is laid out and all the features that it puts out there but I have a problem with it bombing out when I install something. I tried installing several packages with it and it seems to just close the main window and shut down with the following error left on that command line:

```
[Tue Aug 14 00:30:55][root localhost][jobs
1][/var/www/articles/a76]$ deity-gtk
Update ID = 0 Status = Connecting to
http.us.debian.org
Update ID = 0 Status = Waiting for file
Update ID = 0 Status =
Start download ID 1: http://http.us.debian.org
unstable/main knews 1.0b.1-5 - knews
end download ID 1: http://http.us.debian.org
unstable/main knews 1.0b.1-5 - knews
Selecting previously deselected package knews.
(Reading database ... 48791 files and directories
currently installed.)
Unpacking knews (from .../knews_1.0b.1-5_i386.deb)
...
Setting up knews (1.0b.1-5) ... Package manager
(dpkg) succeeded.
Gdk-ERROR **: X connection to :0.0 broken
(explicit kill or server shutdown).
[Tue Aug 14 00:33:08][root localhost][jobs 1]
[/var/www/articles/a76]$
```

Basically, it succeeds in installing the package but it bombs out after it is finished. I'd like it to stick around a little longer. But to be honest, the tool has some very nice things going for it, such as extensive configuration ability in the area of package presentations, something that can be very useful. Another thing I like about it is it is pretty much the same thing as deity-curses, it works the same way. By double clicking a package, it will list the packages that it requires to be installed so you can see what it will do. When you actually click on the "install button", it will show you a list in a new window of what packages it will install. For example, as you can see in the following screenshot, I wanted to install "ari-yahoo" and "balsa". Of course, it adds in the packages that will be required to make this happen without me having to install those dependancies manually. This is the power of having Apt as the back end.



Once I click on the green light to complete the install, Deity-gtk pops up another window which shows me a progress report of what package is currently being downloaded so I can follow the progress of the packages as they are downloaded. The only part that could use some improvement is what happens next. Basically, things get handed over to Apt apparently and we see the following on the console where we ran the program from:

```
.
.
. Selecting previously deselected package ari-
yahoo.
```

```
(Reading database ... 48803 files and directories
currently installed.)
Unpacking ari-yahoo (from .../ari-yahoo_1.8-
1_i386.deb) ...
Selecting previously deselected package libltdl3.
Unpacking libltdl3 (from .../libltdl3_1.4b-
3_i386.deb) ...
Selecting previously deselected package libesmtp5.
Unpacking libesmtp5 (from .../libesmtp5_0.8.3-
1_i386.deb) ...
Selecting previously deselected package
libpspell2.
Unpacking libpspell2 (from .../libpspell2_0.11.2-
2.5_i386.deb) ...
Selecting previously deselected package gnome-pim.
Unpacking gnome-pim (from .../gnome-pim_1.4.0-
7_i386.deb) ...
Selecting previously deselected package balsa.
Unpacking balsa (from .../balsa_1.1.7-2_i386.deb)
...
Setting up ari-yahoo (1.8-1) ...
Setting up libltdl3 (1.4b-3) ...
Setting up libesmtp5 (0.8.3-1) ...
Setting up libpspell2 (0.11.2-2.5) ...
Setting up gnome-pim (1.4.0-7) ...
Setting up balsa (1.1.7-2) ...
Package manager (dpkg) succeeded.
```

At least the process was successful and guess what? It didn't bomb out this time. I still have the program's main user interface sitting there waiting for me to use it again. But in this day and age, the newbie expects all the stuff to be on the GUI, not on some arcane console somewhere such that if they make an icon on their desktop that if they ran deity-gtk from there, where would those console messages go then? Those all important console messages might not be visible and then how would the user know what was going on? Maybe it's not so simple to try and redirect these messages into the GUI somehow but I think it's important because that part of the process, the unpacking and setting up are important to be receiving some feedback that they are actually happening.

I am being hard on this program here because I think it has great potential and it already is a very nice program to use. I just hope to see it become a little more stable and maybe they can get that searching working better. My search queries sometimes don't seem to work as they should. But for basic useage, the tool is there. I was able to do an update of the package list. For those of you that know Apt, that would be apt-get update but for those of you using deity-gtk that would be "go to File >> Update Avail. Package Lists" and that's about it. You get visual feedback on that process too.

I might note that deity-curses has the same situation happen to it when we do package installations. It basically disappears and shows the user the output of apt, as apt performs its unpacking and setting up. There is nothing wrong with seeing such output, but it would be much nicer if that output could be channeled into user interface of the program as being part of the program.

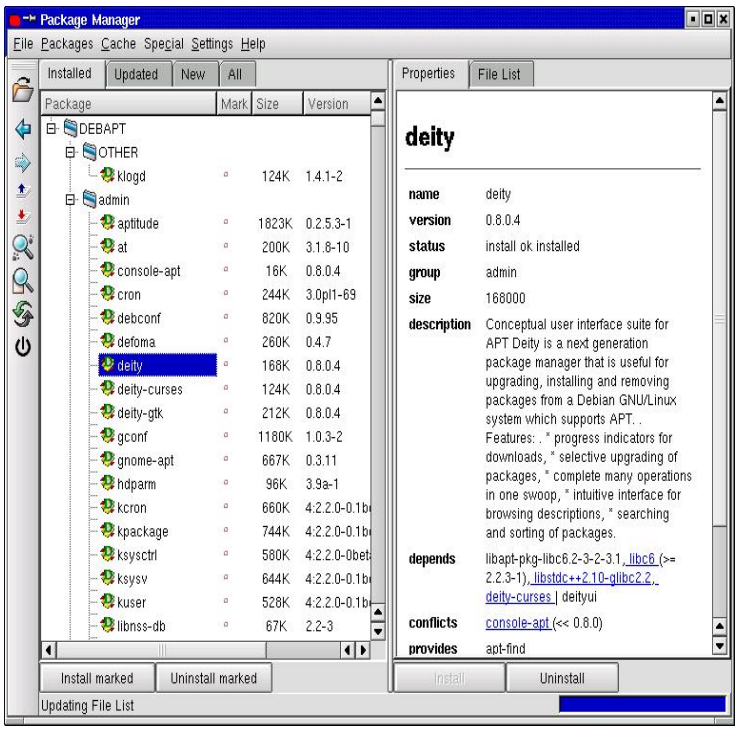
KPACKAGE FROM KDE

Kpackage is an interesting tool since it comes with KDE. I didn't have to go try and download it from a website although I hear it has one at

http://www.general.uwa.edu.au/u/toivo/kpackage/.

If you run KDE, if you go to "K >> System >> Package Manager", you will be running Kpackage. Otherwise, just go to a console or xterm or kterm or whatever command line you use, su to root and type "kpackage" and the program should load. One thing to note though is that if you simply access it from the menu interface, it will be running under your user id and so if you aren't root, then when you try and perform package installations in the program, it seems to pop up a window asking for root's password. That part was kind of confusing because after I typed in the password, it put me at a command prompt in that window. What exactly was I supposed to type then? "exit" ? That's what I typed, but it just sat there.

When I tried running kpackage from a command line as root, things went much better. It worked. Let's take a look at a typical screen from kpackage.



Kpackage has a very nice and intuitive layout, much less complex than that of deity-gtk. Basically the interface is split into two windows side by side. The left-hand window contains lists of packages and the right hand side contains, well, when you select a package on the left-hand window, it shows you a description of that particular package, nicely formatted with all the details that you would like to know about the package, including depends lists, recommends lists and so on. The packages listed in these lists, such as the depends lists are underlined and hyperlinked so that if you click on one of them, the left-hand window switches to the place where the package you clicked on is located so you read about it. The right-hand window shows the description of that package you clicked on too. Wow. This is an amazing feature. I didn't see it in any of the other products. This makes it really easy to investigate package dependancies, for example, to see if you have

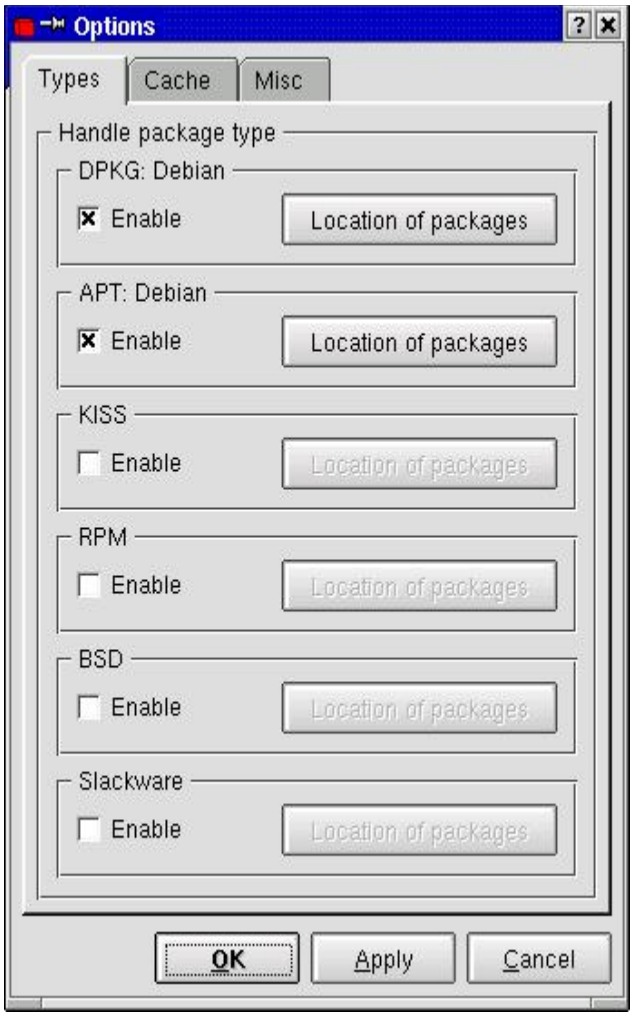
them all met, if you're the type that likes to investigate those types of things. Another useful feature of Kpackage is that it opens up a new window once you click on the "Install marked" button. Then you can see what packages it has planned for install and you click on another button to give it the final go ahead.

Once you do that, you see apt's output in the window to the right, which is part of the window that got popped up. Finally, we have apt integrated into the program window. I like that feature because I think it's important to know not only what packages you are trying to install but what apt actually does install and what it says, for example if there are any errors. I installed eboard, upgraded my kmail by a very small version number and there didn't seem to be any problems.

I didn't stress test the product but from what I can tell, it doesn't really have the tools that would satisfy the power user. Searching and sorting abilities are rather limited, especially the sorting abilities. You basically get what you see and that's it. However, there are 4 panes on the left-hand window that let you see only the list of installed packages, updated packages, new packages or all packages. It's your choice.

The user interface is pretty nice and gets the job done. It's not hard to figure out how to use the program, but you may need to configure it first by going to "Settings >> Configure Kpackage...". It might not be necessary because it seems like it changed the settings on me anyway once I ran the program but here are the settings it is set to now.

Another nice thing about Kpackage is how it uses easily identifiable icons to let you know at a glance whether a package is installed or not. For installed packages, it seems that if you click on the "File list" tab in the right-hand window, with a particular package selected in the left-hand window, you are presented with a list of files that belong to that package. No more typing in dpkg -L package.deb?
The program starts rather sluggishly, not as fast as any of the other programs I tried. I am not sure what the cause of this is but it appears to be related to the fact that it has to parse the package list and sort it a certain way for presentation or something. Hopefully they can work on this aspect of the program, its speed on startup. However, that's a minor issue. The program seems very nice and is a nice addition to the list of programs that we can use to manage our system in Debian. Actually, another plus is that Kpackage can be used to manage rpm-based systems, I believe, so this tool is multi-functional. After all, not everyone running KDE is a Debian GNU/Linux user.
The functions for updating your package list and upgrading your entire system are in the "Special" menu. I knew you were going to ask...



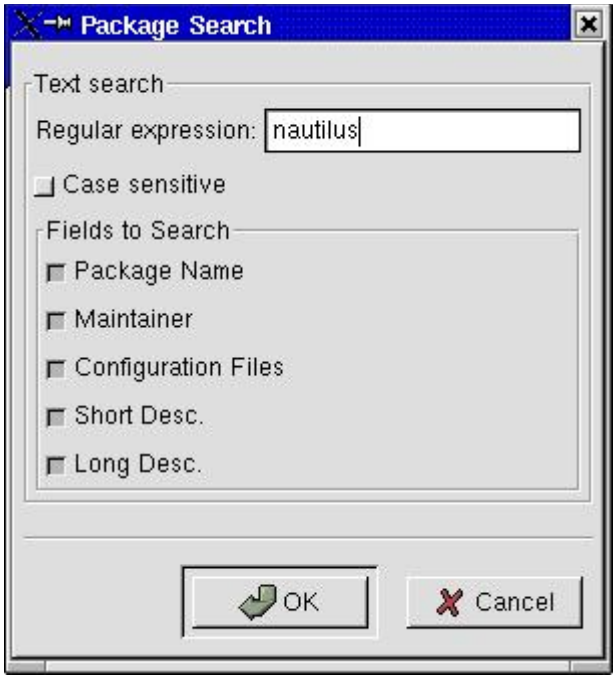
GNOME-APT

In my opinion, Gnome-apt is an extremely useful package management tool. As a graphical user interface-based system, it appeals to people that like to point and click while at the same time it includes some of the power tools that I have come to expect in a package manager. But first let's get things started by looking at the man page for gnome-apt.

```
GNOME-APT(8)
GNOME-APT(8) NAME
       gnome-apt - graphical package management
program SYNOPSIS
       gnome-apt [options] DESCRIPTION
       gnome-apt  is  a  package manager for
Debian with a Gnome front-end.  It provides the
same
       functionality as the apt-get command line
utility.  gnome-apt allows you to easily upgrade
       your  Debian  system,  as  well as install
and uninstall packages using any of the methods
       that apt supports (http, ftp, file).
Eventually gnome-apt will have online help
available
       from within the program. (But right now it
doesn't.) OPTIONS
       Generic Gnome and GTK options can be
accessed using the --help flag.        Help
Options        -?, --help
              Show summary of options.
--usage
              Display brief usage message.
-V, --version
              Show version of program.
Display Options        -g, --geometry
              Set main window geometry in standard
```

```
x++ format. FILES
       /usr/share/gnome-apt/gnome-aptrc
                System-wide gtkrc specific to gnome-
apt (use to configure themes, fonts, etc.)
~/.gnome/gnome-aptrc
                User gtkrc for gnome-apt
/usr/share/gdeb/gdebrc
                Global  gtkrc  with  settings  for
the Details dialog and Package Information pane,
                shared with the currently
unavailable gdeb package viewer.
~/.gnome/gdebrc
                User gtkrc for package information
display         /etc/apt/sources.list
                Configures where apt-get and gnome-
apt look for packages.        SEE ALSO
                apt(8), sources.list(5), apt.conf(5)
AUTHOR
       gnome-apt was written by Havoc Pennington
This manual page was  written  by
       Mitch  Blevins  ,  for the Debian GNU/Linux
system (but may be used by
       others).
GNOME-APT(8)
```

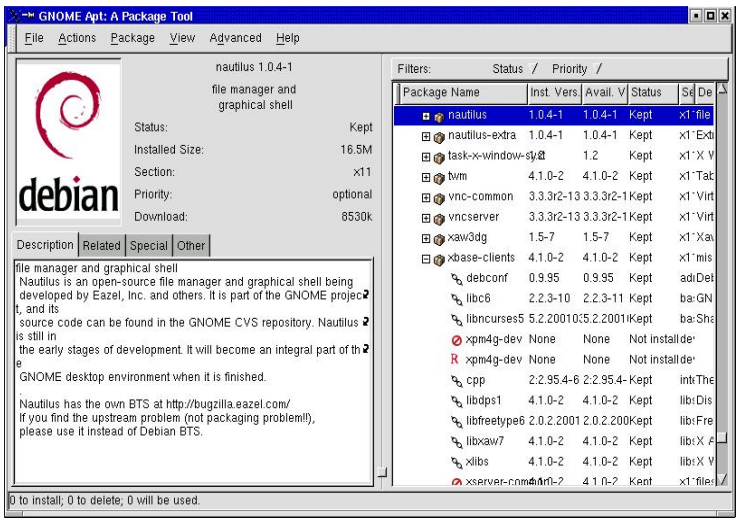
It's simple enough to run it at the command line, as root, with just typing gnome-apt and you will be presented with a nice looking graphical user interface. Now comes the fun part - figuring out how to make things the way you want them, how to configure the tool so that it presents packages in the type of sorted and grouped order that you would like. I'll get back to that later but first I'd like to show you a little about how it interacts with you when you try and install some packages. For example, I wanted to try out Nautilus - Nautilus the file manager. I also wanted to install grub-doc so I could learn a little about grub. That was no problem for gnome-apt. I just searched a little as follows:



And then it popped up a window with a list of matches, any one of which I could click on and it would take me to the package in the main window that matched that listing. Here is what I got back from the search.

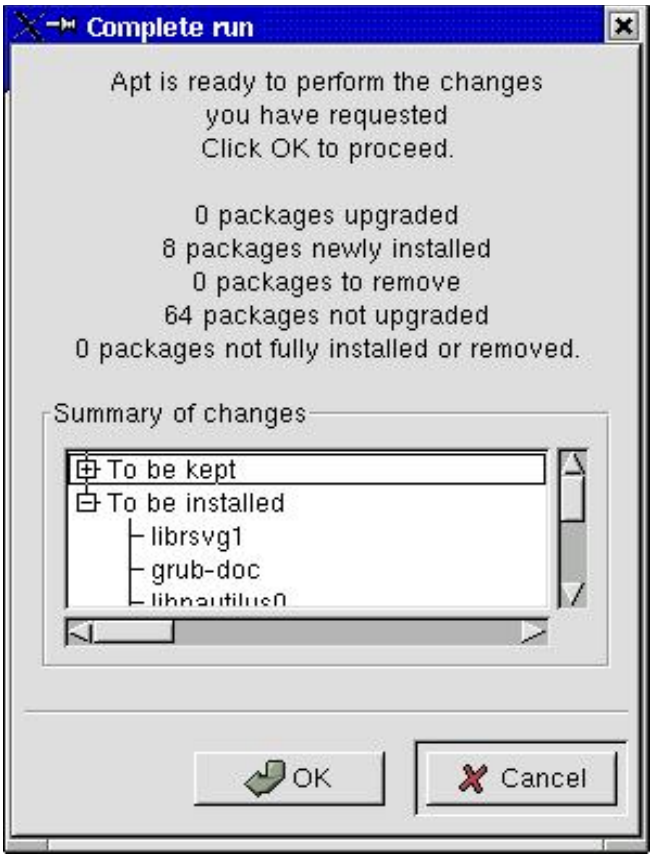
Notice there are even some items that don't even

match the text string that I searched for. The reason for that is that in the search I did I didn't exclude a match that occured in fields other than the Package Name field. But I could have in order to reduce the hits. The only hit that I wanted was one in the Package Name fileld anyway. So I clicked on the match that I knew was the one I wanted.



Note that the above screenshot already shows "nautilus" and "nautilus-extra" as being installed (kept). The reason for that is, I didn't take a screenshot before I installed them but I want to try and keep things in a semblance of sequential order as to what happens. So just bear with me.

So you do the search for nautilus and it pops up a window with a list of matching packages. You click on the one that you think is the right one and you are taken to that package in the main window on the right. Then you simply right-click the package with the right mouse button and select "Install/Upgrade". Select as many different packages as you like because it will not do anything until you go to the **Actions** menu on the main menu and select **Complete Run.** At that time, it will present you with a list of all things it intends to do in a new popup window as follows:

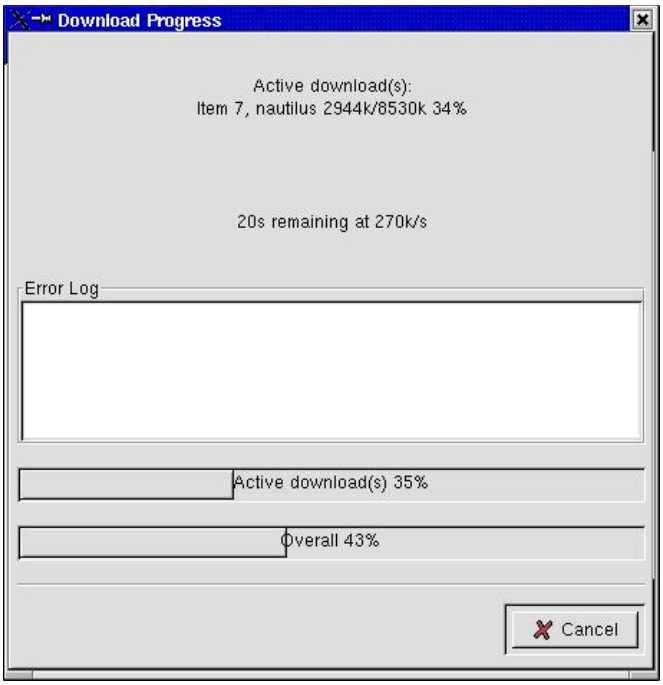


I was presented with two expandable boxes called "To be kept" and "To be installed". By clicking on the "To be installed" check box, it expanded into a list of all the packages it was going to download and install for me, using the apt-get tool to do that. It's important to note that everything is based on the apt-get tool as far as where the packages are retrieved from. My /etc/apt/sources.list file is located here:

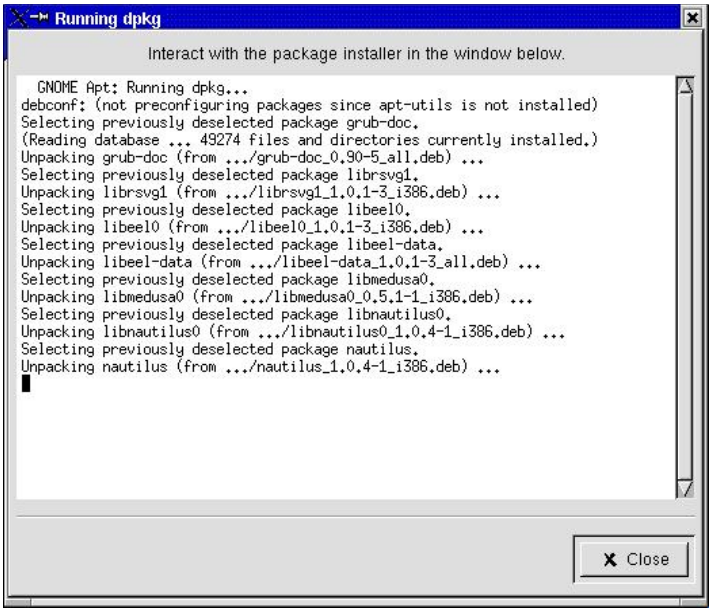
http://www.machineofthemonth.org/articles/a76/sources.txt

if you'd like to read it. It's a very basic one. Very simple. But it works.

So once I agree that this is indeed what I want to do, I can click on "OK" or if I want to back out of it, I can click on the "Cancel" button. Very nice touch there, gnome-apt. Once the user clicks on "OK", what happens is another nice thing. Gnome-apt pops up a new window showing the download progress of each file as it is being downloaded. It shows you how much percentage of the current file has been pulled down, as well as a cumulative percentage of the total sum of the files in terms of file size. This allows you to get a good idea how much longer the process of downloading will take. Very, very nice indeed. I have to say, this program impressed me.



But there's more! Once the packages are downloaded, all 8 or however many of them there were, we aren't done yet. Well, technically you and I were done when we gave our final OK but the program still needs to perform unpacking and setting up of the packages that it just downloaded. It hands that off to maybe the dpkg program or some base debian packaging program behind the scenes. And we see a new screen come up that looks like the following.



That screen continues to produce output until it shows that the unpacking and setting up has been completed succesfully and we can then click the "Close" button to close that window and go back to our main program window and begin installing more packages, or what have you. Did you notice the way that everything is integrated into the program? The output from all of the downloading, unpacking and setting up is contained within the program. This is

very nice and it is what some package management programs are lacking, as we have discussed earlier.

Another nice feature of gnome-apt that has got to be appreciated is how it uses symbols to identify various levels of relationships among packages. If you look back at the screenshot of the main gnome-apt program window, you can see that I have double clicked on the "xbase-clients" package, just to expand it so we can see it's "dependancies". That's right. When you double click on the box to the left of the package name you get an expanded list of packages that are related to it in some way. In reality, let's take a look at the output of the following command:

```
[Wed Aug 15 03:25:28][glenn_m localhost][jobs
3][/var/www/articles/a76]$ dpkg -s xbase-clients
Package: xbase-clients
Status: install ok installed
Priority: optional
Section: x11
Installed-Size: 3888
Maintainer: Branden Robinson Source: xfree86
Version: 4.1.0-2
Replaces: xbase (<< 3.3.2.3a-2), xf86setup (<< 3.3.2.3a-9),
xserver-common (<< 4.0), xmodmap, xcontrib, xpm4g-dev, xpm-bin, xsm
Provides: xmodmap, xcontrib, xpm-bin, xsm
Depends: debconf (>= 0.3.83), cpp, libc6 (>= 2.2.3-7), libdps1 (>>
4.1.0), libfreetype6, libgl1, libncurses5 (>= 5.2.20010310-1),
libxaw7 (>> 4.1.0), xlibs (>> 4.1.0)
Conflicts: xbase (<< 3.3.2.3a-2), xserver-common (<< 3.3.2.3a-9),
xmodmap, xaw-wrappers (<< 0.90), xfonts-100dpi (<< 3.3.3.1-3),
xfonts-75dpi (<< 3.3.3.1-3), xfonts-base (<< 3.3.3.1-3), xfonts-
cyrillic (<< 3.3.3.1-3), xfonts-scalable (<< 3.3.3.1-3), xfnt100
(<= 3.3.2.3a-1), xfnt75 (<= 3.3.2.3a-1), xfntbase (<= 3.3.2.3a-1),
xfntcyr (<= 3.3.2.3a-1), xfntscl (<= 3.3.2.3a-1), xdm (<< 4.0),
xsm, xcontrib, xpm4g-dev, xpm-bin
Conffiles:
```

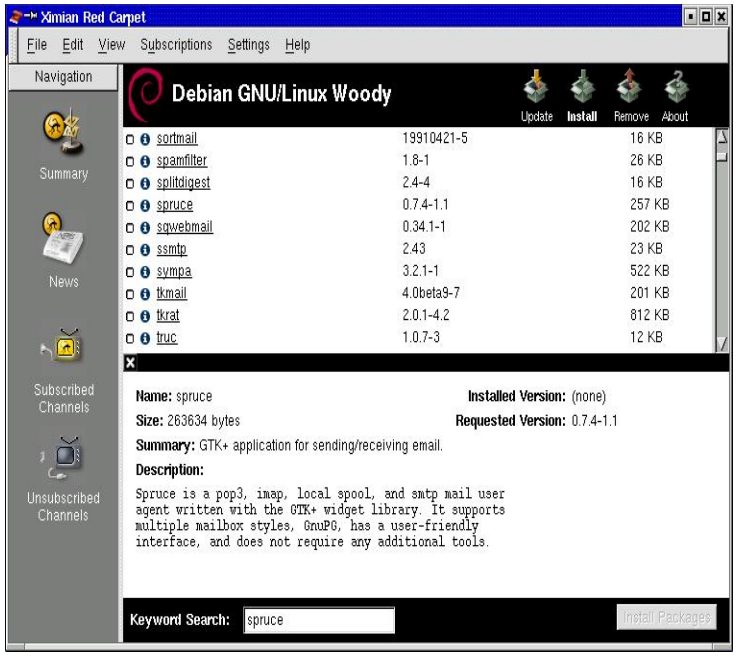
It goes on and on but the point is that Debian includes package dependancy stuff g with every package so that programs like gnome-apt are free to use that information to present very nice displays to the user. If you compare the graphical symbols next to the dependancies listed under "xbase-clients" in the main window to the dependancies listed above in the output of the above listing, you will see that the chain symbol means "depends". For example, look at cpp or libdps1 or debconf or libc6 in the main window under the xbase-clients listing. To the left of each one of them is an icon that looks like a chain. Look at the output of dpkg -s xbase-clients and we see that the above packages all happen to Depend upon our xbase-clients package. What about that red R symbol? As it turns out, it means "Replaces". For example, you can't see it in the window because it is on down further in the list but xpm-bin has a red R symbol to the left of it, under xbase-clients. So what does that mean? We look in the above output once again to find where xpm-bin falls. It actually falls under two places, Replaces and Provides but the reason xbase-clients provides xpm-bin is because it replaces it so that is what the R stands for. There are some other symbols too. For example, the red circle with a diagonal line in it. That stands for Conflicts. It means that the two packages conflict with one another and cannot be installed at the same time. There are some other symbols you will see too, such as a hand with a thumb raised. And you will also see a yellow pyramid with a black exclamation mark. I'll let you figure those out.

Updating your packages list is simple enough - just go to "Actions >> Update" and you get the good type of feedback we have been seeing all along with gnome-apt. Question: How do I perform an apt-get dist-upgrade? I don't see a menu element to do that. Hmmm. Maybe that is done using some combination I am not aware of.

The ordering and grouping and searching are awesome. The filtering capabilities are also fairly extensive although I am not sure they work 100%. The ordering and grouping are accessed through the "View" menu and let you do some fairly sophistocated package presentation in the main window so you can get down to business. The filtering is done in the right hand window using the drop down boxes at the top.

Ximian Red Carpet

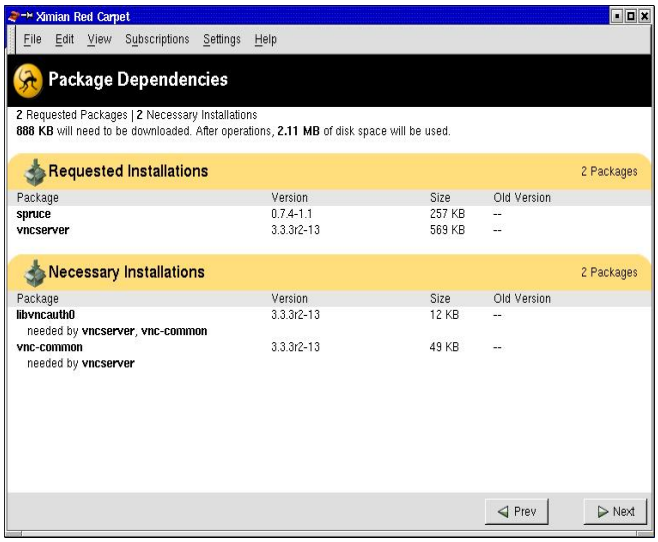
I installed the spruce email client:
http://spruce.sourceforge.net/
and vncserver program using Ximian's Red Carpet tool. Basically, my impression of Red Carpet is favorable while at the same time realizing that it isn't the power user's dream come true. However, the program does have it's nice points. One of those is that it appears to be fairly stable and contains the entire operations within its GUI, which is always a good thing. When you run the program from the command line, it seems to connect to a site somewhere and pull down an updated list of packages if there is one. Sometimes it doesn't do that but sometimes it does. I am not sure how much data is being transferred either.



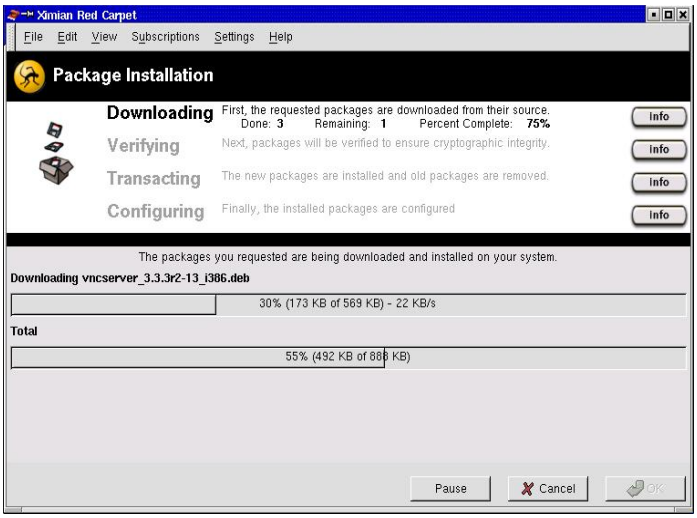
Red carpet is run as root and you just type red-carpet at the command line. By clicking on the blue information icon next to a given package, the window splits in to a top and bottom and you can read about the package you clicked next to. This is nice. There are also minimal searching features that allow the user to find a package containing a string. There doesn't appear to be any way to re-arrange the grouping of the packages, although for normal everyday users, this probably would not present any great problems. It's not the best situation but I can

understand why Ximian may not have wanted to delve too deeply into grouping and sorting and filtering. Those are complex things and I think they want to appeal to simplicity and power through that simplicity and I think they succeeded in that. The categories appear in black with a yellow background and package names are alphabetical within each category. This seems reasonable.

To install a package, you just click the box next to its name. Click 1 or 100 boxes and when you are finally ready to begin the installation process, you simply click the "Install Packages" button at the lower right corner. Then Red Carpet presents you with some feedback showing what it needs to do to make things happen. You can either accept its recommendation and proceed or go back and re-select packages. I clicked on the "Next" button.



The next screen that pops up takes us through the actual download, verify, transacting and configuring processes. We receive on-screen feedback at each step of those processes. What a nice user interface. Ximian really got this part right.



Basically, there is only one Debian channel, which is called "Debian GNU/Linux Woody". Am I to infer that this is the testing branch at Debian? What if I wanted to be on the cutting edge and run the unstable branch.

(For a description of the differences, go to http://www.debian.org/distrib/packages.) How would Red Carpet get along with the rest of my system or am I suppsed to use Red Carpet soley and not use any of the other tools that this article discussed. The other tools all seem to share /etc/apt/sources.list. So they get along. Red Carpet I am not so sure about it though. Part of the attractiveness of tools like deity and dselect and gnome-apt and kpackage is that they all use the same base tools, apt-get, the sources listed in /etc/apt/sources.list so that when for example, a package list gets updated, it doesn't matter if I run gnome-apt next time because gnome-apt will see those changes. Unfortunately, it appears that Red Carpet does its own internal updating of package lists and keeps them in a different location. But this doesn't mean that it can't be used side by side with stuff like deity-curses or aptitude. I suppose it can but I would like to know a little more about Red Carpet and how the packages it installs would affect the packages that I try and install using one of the other tools, one of the tools that uses my sources.list sources for pulling down stuff. It looks like Red Carpet is only providing me the main section and not contrib or non-free or non-US. Well, that's not good either because Linux users deserve choice. I need to be able to choose what sections I want to have available for downloading and installing onto my system.

However. Red Carpet is a fascinating little program and does relatively well, at least in my testing. It is another good reason to use Debian. It is interesting to note that Red Carpet, like Kpackage, is not a Debian-only package management tool. It supports other package formats and other linux distros too. It has a very polished feel to it and Ximian should be commended for their good work.

As for a small analysis, it stores the package lists (packages.gz) in /var/cache/redcarpet/ as we can see:

```
[Mon Aug 13 23:50:35][root localhost][jobs 0][/home/glenn_m]$ ls -l
/var/cache/redcarpet/
total 1381
-rw-r--r--    1 root     root         1814 Aug 13 15:21
channels.xml.gz
-rw-r--r--    1 root     root      1319234 Aug 13 22:38 debian-
dists-woody-main-binary-i386-Packages.gz
-rw-r--r--    1 root     root         2753 Aug 13 23:24
distributions-common-debian-channel.png
-rw-r--r--    1 root     root         1324 Aug 13 22:52
distributions-common-debian-subs.html
-rw-r--r--    1 root     root         1456 Aug 13 15:23
distributions-common-debian-unsubs.html
-rw-r--r--    1 root     root         4425 Aug 13 23:24 evolution-
debian-woody-i386--..-common-channel-evolution.png
-rw-r--r--    1 root     root          837 Aug 13 15:21 evolution-
debian-woody-i386--packageinfo.xml.gz
-rw-r--r--    1 root     root         4530 Aug 13 15:23 icons-
evolution.png
drwxr-xr-x    2 root     root         1024 Aug 13 23:41 packages
-rw-r--r--    1 root     root         3716 Aug 13 15:21
prettynames.xml.gz
-rw-r--r--    1 root     root          988 Aug 13 15:21 red-
carpet.rdf
-rw-r--r--    1 root     root         4315 Aug 13 23:24 ximian-
gnome-debian-woody-i386--..-common-channel-ximian_gnome.png
-rw-r--r--    1 root     root        54859 Aug 13 15:21 ximian-
gnome-debian-woody-i386--packageinfo.xml.gz It puts the deb
packages into its own directory also. What this all means and what
```

```
implications it has for the other package management tools, I am
not sure. It's just something I noticed.
[Mon Aug 13 23:48:37][root localhost][jobs 0][/home/glenn_m]$ ls -l
/var/cache/redcarpet/packages/
total 983
-rw-r--r--    1 root     root        85416 Aug 13 15:31
kterm_6.2.0-35_i386.deb
-rw-r--r--    1 root     root        12872 Aug 13 23:40
libvncauth0_3.3.3r2-13_i386.deb
-rw-r--r--    1 root     root       263634 Aug 13 23:40
spruce_0.7.4-1.1_i386.deb
-rw-r--r--    1 root     root        50252 Aug 13 23:40 vnc-
common_3.3.3r2-13_i386.deb
-rw-r--r--    1 root     root       583364 Aug 13 23:41
vncserver_3.3.3r2-13_i386.deb
```

Conclusions

The tools that we have taken a look at in this article are tools that I intend to keep on my system. I think each one of them offers something valuable and I want to experience their value. Package management is not one of the sexy applications of computer technology but it has become a very important one. For example, I don't think I would have been able to write this article had it not been for the aptitude package manager downloading and installing Ximian's Red Carpet and all of its dependancies. Trying to do all that by hand or compile stuff from source would have taken a great deal of time and research and effort. Not that it would have been not worth it but sometimes you just want to give something a test run and see what it's like. Now that I know how nice some of these tools are, maybe I might want to do that in the future, do it by hand.

Package management has reached a fairly mature point in debian. With the exception of Red Carpet, it appears that all of the package managers we saw in this article can be configured to use the /etc/apt/sources.list file and hence, let you and me configure where and what we want to have access to. That can only be a good thing. Part of the difficulty with using Linux is managing your time. To manage your time, you have to have an efficient way to scout through a listing of all the software you could install and all the software you already have installed. You should be able to delete already installed software and add new software easily with the press of a few buttons. Debian package management is at this level.

You and I are the ones that will benefit from this. If I missed anyone's favorite package manager, please do let me be aware of it.

Resources

Dpkg/dselect is located at

http://packages.debian.org/unstable/base/dpkg.html

Gnome-apt can be found at
http://packages.debian.org/unstable/admin/gnome-apt.html

Aptitude is located at
http://packages.debian.org/unstable/admin/aptitude.html

Kpackage is linked to at Freshmeat at
http://freshmeat.net/projects/kpackage/

but the site it links to doesn't appear to have a download option. As I pointed out earlier though, you can get Kpackage as part of KDE now so I guess you don't need to worry about this if you have a KDE install.

Deity is located at
http://packages.debian.org/unstable/admin/deity.html

and there are links to Deity-curses and Deity-gtk from that page.

Red Carpet is a Ximian product and can be found at the Ximian website at
http://ximian.com/apps/redcarpet.php3

I would be very remiss not to mention Apt, which is located at

http://packages.debian.org/unstable/base/apt.html.

Even though we didn't directly look at apt, it's was there behind the scenes. Almost all of the packages we reviewed depend upon it!

This article is re-printed with permission. The originals can be found at:

http://www.machineofthemonth.org/articles/a76/index.html

Learning with nmap

Author: Danilo Lujambio <danilo@tau.org.ar>

Abstract

Why are scanners so important for the security of networks? Basically because they are essential tools for those who want to attack a system. The preparation of an attack by a cracker could look as follows:

- Scan a target machine or selected network, observe which services are offered and which operating systems runs these services, and work on some well-known vulnerability in any of them.
- Scan any network or machine, look for a service or operating system (including the checkup of the version) with a known vulnerability.

For a system administrator who is aware of system security, it is important to carry out a scanning of their own network, and look for vulnerabilities before others do it with not so good intentions.

There are several scanning tools for this purpose, but the article will only look at nmap. Nmap is among the most complete scanners and security tools.

Nmap allows the system administrator to scan the networks in order to know which servers are active and which services they offer. For this purpose, nmap offers several scanning techniques. This article will work on a limited number of them, reviewing (maybe teaching?) some aspects of TCP protocol

The strategy in this article will be to show some of the more common ways to use nmap, to obtain information about systems and, in parallel, show how to find traces of scanning on the target side.

The nmap can be obtained from www.insecure.org. After downloading run:

```
tar zxvf nmap-2.30BETA17.tgz
cd ...../nmap-2.30BETA17/
./configure
make
make install
```

and it is installed.

The nmap output is usually a list of "interesting" (active)
ports on the scanned target machine. These ports provide you with the name of the service, the state and the protocol.

Scanning with TCP, three way handshake of TCP (option -sT)

The simpler form of nmap scanning is done with option -sT. It is based on the method of establishing a connection in the TCP protocol, known as a three way handshake. The sequence [1] is roughly described below:

1. The server must be ready to receive a connection (usually using the socket, bind and listen functions)
2. The client starts an active connection - a call to `connect ()`. This sends a SYN segment to the server to inform about the initial sequence number of the data that client will send during connection. The SYN usually contains an IP Header - a TCP Header and maybe some TCP option.
3. The server should acknowledge the SYN sending with an ACK and a SYN with its sequence number (within the same TCP package).
4. The client should acknowledge the server SYN with an ACK

This way of scanning has two advantages:

- it is fast (nmap even has options that we will not analyze to make it faster on slow connections)
- special privileges are not needed on the machine that launches the scanning but it has a big disadvantage. It is very simple to detect and easy to filter.

We will follow the procedure used by nmap option -sT, running tcpdump in the target machine. nmap is executed on machine 192.168.255.20 and points toward machine house2.xxx.xxx.xxx, through an Ethernet network.

```
1) 08:24:18.393108
        192.168.255.20.1024 house2.xxx.xxx.xxx.653: S
        2632227152:2632227152(0) win 16060 < mss
        1460,sackOK,timestamp 232602[|tcp] (DF)
2) 08:24:18.393167 house2.xxx.xxx.xxx.653 192.168.255.20.1024:
        R 0:0(0) ack 2632227153 win 0
3) 08:24:18.393227 192.168.255.20.1025
        house2.xxx.xxx.xxx.6141: S 2644226118:2644226118(0) win
16060
        < mss 1460,sackOK,timestamp 232602[|tcp] (DF)
4) 08:24:18.393258 house2.xxx.xxx.xxx.6141
        192.168.255.20.1025: R 0:0(0) ack 2644226119 win 0
5) 08:24:18.453343 192.168.255.20.1298
        house2.xxx.xxx.xxx.pop3: S 2640612362:2640612362(0) win
16060
        < mss 1460,sackOK,timestamp 232608[|tcp] (DF)
6) 08:24:18.453542 house2.xxx.xxx.xxx.pop3
        192.168.255.20.1298: S 1658259980:1658259980(0) ack
2640612363
        win 16060 < mss 1460,sackOK,timestamp 243353[|tcp] (DF)
7) 08:24:18.458667 192.168.255.20.1298
        house2.xxx.xxx.xxx.pop3:. ack 1 win16060<nop,nop,timestamp
        232609 243353 (DF)
8) 08:24:18.461280 192.168.255.20.1298
        house2.xxx.xxx.xxx.pop3: F 1:1(0) ack 1 win 16060 <
        nop,nop,timestamp 232609 243353 (DF)
```

Line numbering was added to ease the explanation. Line 1 shows the "attacking" machine 192.168.255.20 sending a SYN segment from port 1024 to port 653 on the target machine house2.xxx.xxx.xxx. We can recognize it as a SYN segment thanks to the S after the port number (653). This covers the point 2) of the three way handshake as explained above. In line 2 we see the target machine responding with a RESET package (notice the R after the 1024) indicating that there is no process listening on port 653. Lines 3 and 4 are similar to the first ones, but checking if there is a process on port 6141 of target machine. As there is nothing again, it also answers with a RESET Line 5 shows how the machine 192.168.255.20 sends a SYN segment to POP3 port of target machine (number 110), and the target machine answers with an ACK accepting the SYN and the sequence number (It sends the sequence number from target machine,

1658259980 in this case, and the sequence number sent by the host 192.168.255.20, adding 1, that is 2640612363). Notice that the packet sent from house2 has the control bits SYN and ACK activated. This is seen in line 6 and is the step 3) of the three way handshake above. Line 7 shows the recognition of the last packet received on host 192.168.255.20 with an ACK segment, reaching the point 4) of the handshake. Line 8 is the connection closing from 192.168.255.20, which is done sending a FIN segment (notice the F after pop3)

This run allowed nmap to detect that port 110 (pop3) of house2 is an active one on this machine.

As stated above, this way of scanning is easy to detect, using the prints left in file /var/log/messages (although this depends on the way that syslog.conf was configured) the connection seen in lines 5 to 8 produced:

```
May 6 08:24:01 house2 in.pop3d[205]: connect
from root@192.168.255.20
```

Scanning using SYN segments (half open, option -sS)

This scanning type is performed by executing nmap with option -sS. The technique used is to open a "half connection": we send a SYN segment and, if an ACK is received then we have detected an active port on the target machine, and we sent a RESET to close the connection promptly. If we receive an RST instead of an ACK, then the scanned port is not active. This scanning procedure has the drawback that root privileges are needed to execute it. But it has the advantage that is difficult to detect in the scanned machine.

Let's see a similar analysis of the actions done by nmap with this option, analyzing it with tcpdump (with lines numbered again for easier description)

```
1) 22:25:45.856936 192.168.255.20.40175
          house2.tau.org.ar.946: S 1292785825:1292785825(0) win
          3072
2) 22:25:45.857078 house2.tau.org.ar.946
          192.168.255.20.40175: R 0:0(0) ack 1292785826 win 0
```

Lines 1 and 2 are quite close to lines 1 and 2 of the previous section, except that a SYN segment is seen, sent by host 192.168.255.20 to port 946 of host house2 and we get the answer with a RESET because it is not an active port.

```
3) 22:25:45.970365
        192.168.255.20.40175 house2.tau.org.ar.pop3: S
        1292785825:1292785825(0) win 3072
4) 22:25:45.976022 house2.tau.org.ar.pop3
        192.168.255.20.40175: S 185944428:185944428(0) ack
1292785826
        win 16080 < mss 536 (DF)
5) 22:25:45.979578 192.168.255.20.40175
        house2.tau.org.ar.pop3: R 1292785826:1292785826(0) win 0
```

Lines 3, 4 and 5 are produced by the successful discovery of a service (pop3) at port 110 from host house2. As mentioned, the three way handshake it is not completed but when nmap receives the recognition of its SYN segment (by means of the ACK segment sent by house2 in line 4), it sends a RESET segment that forces the communication to interrupt.

This scanning over house2 didn't leave any trace in the file /var/log/messages, as stated earlier.

Scanning using the FIN segments

This scanning is based on the fact that inactive ports on the target machine respond to a FIN package with a RST package. On the other hand, active ports simply ignore those packets. Therefore the list of interesting active ports is obtained by observing which are those that have not answered. Hosts running Microsoft operating systems can not be scanned with this method since they have a non standards-conforming implementation of the TCP protocol.

There are three forms of operation of nmap using similar
techniques, achieved through options -sF, -sX and -sN. We will further analyze the behavior of option -sF, performing an analysis similar to the one for the previous sections.

```
1) 06:50:45.643718 192.168.255.20.35600
        casahouse.tau.org.ar.864: F 0:0(0) win 2048
2) 06:50:45.643865 house2.tau.org.ar.864
        192.168.255.20.35600: R 0:0(0) ack 0 win 0
```

In lines 1 and 2 the FIN segment delivery (notice the F after the 864 in line 1) is observed on the target host, which answers with a RST packet (notice the R in line 2 after the 35600). nmap concludes that the 864 in house2 is not active

.

```
3) 06:50:47.933227 192.168.255.20.35600 >
          house2.tau.org.ar.pop3: F 0:0(0) win 2048
4) 06:50:48.251147 192.168.255.20.35601 >
          house2.tau.org.ar.pop3: F 0:0(0) win 2048
```

Lines 3 and 4 take the pop3 port on house2 as an example. In line 3 we see a FIN segment sent, which doesn't get an answer from house2. Line 4 was a surprise, being probably a measure taken by nmap to check the status of that port, sending another FIN segment to ensure that port is not answering. In both cases, house2 ignored the packets, showing to nmap that port pop3 is active.

Functionality of tcpdump that can help

In the section about "the three way handshake scanning" you saw the traces that a scan can leave behind, and in the later sections, you saw the scanning with options -sS and -sF that don't leave any footprints. We can use tcpdump to detect this type of scanning on a host connected to a network which could be a target of attacks. The drawback of tcpdump is that it generates an enormous amount of information and raises difficulties with regards to storage and analysis. Some expressions are shown here that act like filters, such that the information obtained is smaller and simpler to analyze. To make it easier to understand the expressions, we show below the format of a TCP packet [2].

TCP (RFC 793)

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgement Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Offset |  Reserver |U|A|P|R|S|F|             Window            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

We see that the 13th byte is the one where the flags resides which identifies the kind of packet (SYN, FIN, etc.). With this knowledge and the and (&) operator we can construct masks to detect the active bits, building expressions such as

```
tcpdump ' tcp[13] & 7 != 0 and dst 192.168.255.20
' > /tmp/out7
```

which filters the input leaving the packets with bits R, S or F active (the mask is 00000111) with 192.168.255.20 as destination host (obviously this number IP will be looked at by the machine)

Using

```
tcpdump ' tcp[13] & 1 != 0 and dst 192.168.255.20
' > /tmp/out1
```

we will obtain the packets with an active FIN bit (the mask is 00000001). It can be useful to detect the nmap scanning with option -sF. And with

```
tcpdump ' tcp[13] & 2 != 0 and dst 192.168.255.20
' > /tmp/out2
```

we will get only the packets with an active SYN bit, being useful to detect scans with option -sS

For the last type described (with option -sS) there are specific detection programs available [3].

Conclusion

Programs such as nmap are very useful to improve the system security by looking at networks through the eyes of a potential cracker. We have shown the operation of a rather small part of the options, but hope it helps you to understand the idea of network scanners a bit more.

Bibliography

[1] W. Richard Stevens Unix Network Programming Volume 1
[2] RFC 793
[3] to see nmap documentation

This article is re-printed with permission. The originals can be found at:

http://linuxfocus.org/English/July2001/article170.shtml

The Open Source Lucky Dip

By: Con Zymaris <conz@cyber.com.au>

Welcome back.

OSD makes a return this issue (mostly, because we have enough room to put it in!) As always, a pot-pouri of apps and utils are included. If you find that you have a burning desire to see your favourite (but obscure) utility in lights, post me a note and a URL. auugn@auug.org.au

❍ ❍ ❍

Affix: Reduce Bluetooth protocol development aches

I've been involved in at least one project this year which included Bluetooth technology running on Linux for a wireless devices. Maybe you have too? If so, Affix may have save you a lot of time and money. Affix is an implementation of a collection of Bluetooth functionalities for Linux 2.4.x kernels. It supports core Bluetooth protocols like L2CAP, RFCOMM, SDP, and different Bluetooth profiles like Serial Port. Affix features modular implementation, a socket interface to L2CAP and RFCOMM protocols, Bluetooth module interface independence, and multiple Bluetooth devices support. Developed by one of the leading exponents of Bluetooth, the Nokia Research Center, you can guess that this protocol stack is up to scratch. http://www-nrc.nokia.com/affix/

Atlantik: Network Monopoly®

Quick, someone call the DoJ! It's another Monopoly, but hey, this one is just in time for those more financially opportune members of the family for the swelter-season. Atlantik is a KDE client for playing the Atlantic and Monopoly boardgames with a monopd/atlanticd server. ***<insert pithy quip>*** You don't need to be a Bill Gates to win in this monopoly game...but hey, perhaps you could de-commoditise the game network protocols? ***</insert pithy quip>***
http://capsi.com/atlantik/

Yet Another System Information Script

From the 'never-can-have-too-much-information-about-your-servers-school-of-thought' comes YASIS. YASIS is a Perl port of phpSysInfo which, when accessed from a browser, collects various statistics from your Web server and outputs the result in a nicely-formatted HTML page. You can grab a copy from http://yasis.sourceforge.net

XIWA: XIWA Is Web Accounting

Well, for someone who believes that you can do almost anything with web-based applications, this one surprised even me ;-) XIWA is a Web-based accounting package built with Perl and PostgreSQL. It supports Double Entry/Stocks and has a powerful, flexible reporting engine. Written by James A. Pattie, can fetch a copy (it's GPL) from http://xiwa.sf.net/

Wordtrans: Ich Bin ein Translator

Wordtrans is a frontend for several dictionaries. It supports some plain text dictionaries such as i2e (English-Spanish) and de-en (German-English), Babylon Translator dictionaries, and dict servers dictionaries. Some features include console and X (Qt) versions, good speed, and the ability to watch the clipboard and automatically translate the word there. Download it from:
http://wordtrans.sourceforge.net/index.php

XCMS: Content Management System

Developed by Felix Rabe, the Xitnalta Content Management System (xcms) provides a virtual filesystem library to deal with content from different sources (a real filesystem, a database, etc.) and in different formats. Its goal is to provide yet another way to access and manage data both locally and over. XCMS's homepage is at http://xcms.sourceforge.net/

wxBasic Programming Language

If you had a hankering for the good 'ole simple days of BASIC, check this project out. Written by David Cuny, wxBasic Programming Language is a cross-platform Basic interpreter. It combines the simplicity of Basic, with the portability of the cross-platform wxWindows library. Download it from:
http://wxbasic.sourceforge.net/

The Gallery

According to the blurb, Gallery is a slick Web-based photo album written using PHP. It is easy to install, includes a config wizard, and provides users with the ability to create and maintain their own albums in the album collection via an intuitive Web interface. Photo management includes automatic thumbnail creation, image resizing, rotation, ordering, captioning and more. Albums can have read, write, and caption permissions per individual authenticated user for an additional level of privacy. Gallery was written by Bharat Mediratta. It's GPL and can be had from:
http://gallery.sourceforge.net/

SmsSend: SMS-me-*baby!*

In case you want to setup a system whereby you batch-jobs or your network monitoring tools can hurl short text notes at your phone across the aether, consider SmsSend. SmsSend allows you to send free SMS to any GSM, connecting to Internet sites using scripts. It is available both for Windows and Unix. Get the tarball here:
http://zekiller.skytech.org/smssend_menu_en.html

or the RPM here:

http://www.barsnick.net/sw/smssend.html

Roadsend PHP SiteManager

I know this will prove to be popular for many web-monkeys. SiteManager is an Open Source Web Application Toolkit for PHP Developers that includes a framework for code modules and layout templates, database connectivity, SmartForms, sessions, and some other tools. All parts of the system are integrated through object-oriented libraries which allow for easy expandability and maintenance. Home is here: http://www.roadsend.com/siteManager/

IzPack: Java Installer

I can imagine many Java coders will like the sound of this tool IzPack is a powerful Java installer builder. It is able to create lightweight and modular installers. You have the choice of the installer panels you want to use (some can do the same job, so that you can select the one you prefer), and you even have the choice of the kind of installer that you want to use. IzPack doesn't use any portion of native code, it is designed to be fully independent from the operating system that runs it. It is very easy for the end user with a properly installed JVM to use an installer made with IzPack, since a single "java -jar installer.jar" will launch it. It's GPL and available from:
http://www.izforge.com/izpack/

Oasis Media Streamer

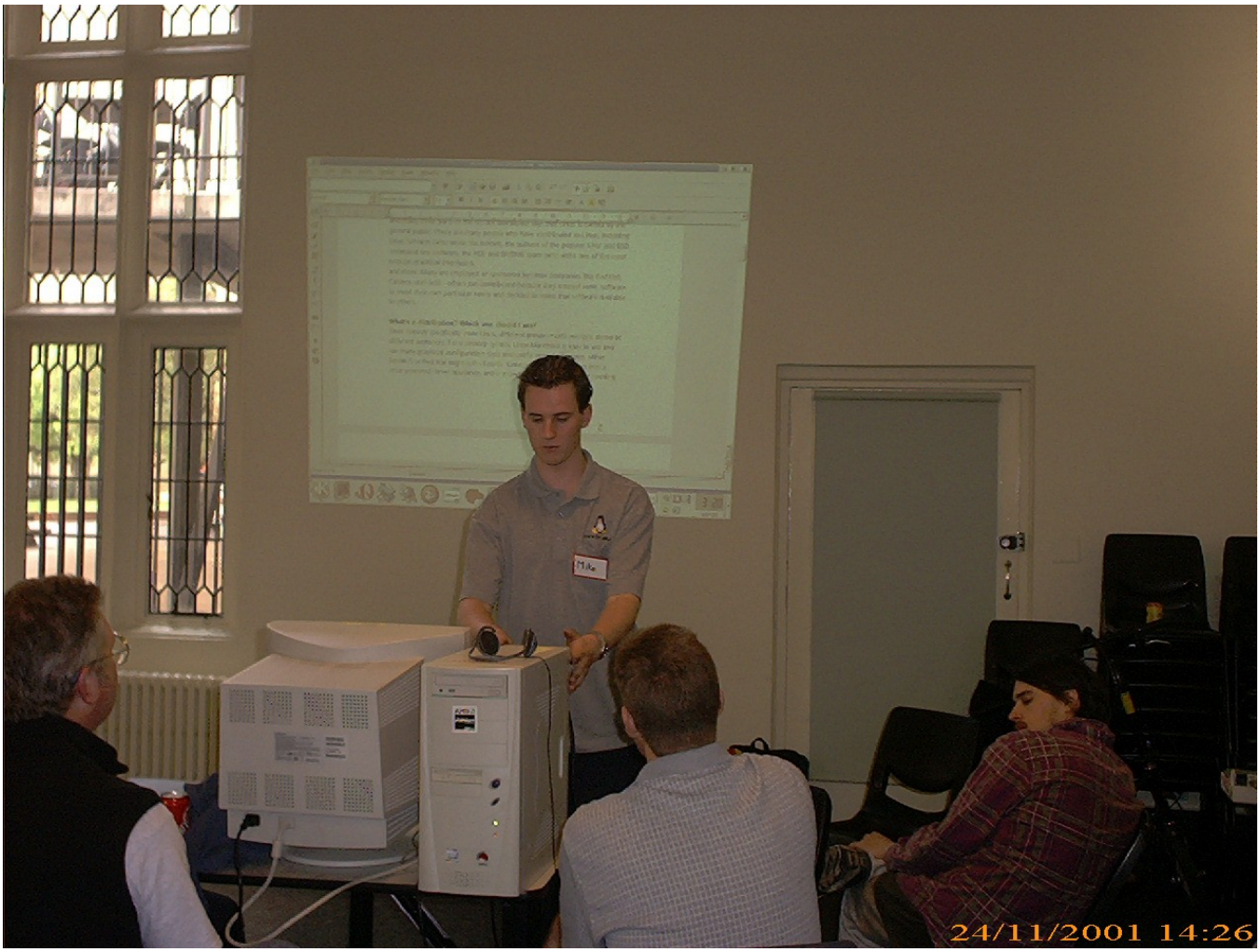
Hey, time to stream the merry sounds emanating from your office's Christmass party ;-) Oasis Media Streamer is a CGI that streams mp3s, videos and images from a Unix webserver. The files and directories are presented in a filemanager-type tree display. Optionally, mp3 files can also be played on the server, rather than streaming them. It is written in C and there are no special library or server requriements. Download from:
http://startuplinux.com/index.html

Advertisement: American Bookstore

LUV: Linux Users Victoria. Installfest2001.

Photographer: Barry Klien

LUV had their annual installfest on Saturday, November the 27th, at Melbourne University. Here are some pictures of the event, showing too many unix-heads guzzling Coke. (settle, I said guzzling.) If you look closely, you can see a few AUUG regulars, sometimes wearing red fedoras. A rollicking time was had by all.


Mike
24/11/2001 14:26

AUUG Chapter Meetings and Contact Details

CITY	LOCATION	OTHER
BRISBANE	Inn on the Park 507 Coronation Drive Toowong	For further information, contact the QAUUG Executive Committee via email (qauug-exec@auug.org.au). The techno-logically deprived can contact Rick Stevenson on (07) 5578-8933. To subscribe to the QAUUG announcements mailing list, please send an e-mail message to: <majordomo@auug.org.au> containing the message "subscribe qauug <e-mail address>" in the e-mail body.
CANBERRA	Australian National University	
HOBART	University of Tasmania	
MELBOURNE	Various. For updated information See: http://www.vic.auug.org.au/auugvic/av_meetings.html	The meetings alternate between Technical presentations in the odd numbered months and purely social occasions in the even numbered months. Some attempt is made to fit other AUUG activities into the schedule with minimum disruption.
PERTH	The Victoria League 276 Onslow Road Shenton Park	Meeting commences at 6.15pm
SYDNEY	TBA	

FOR UP-TO-DATE DETAILS ON CHAPTERS AND MEETINGS, INCLUDING THOSE IN ALL OTHER AUSTRALIAN CITIES, PLEASE CHECK THE AUUG WEBSITE AT HTTP://WWW.AUUG.ORG.AU OR CALL THE AUUG OFFICE ON 1-800-625655.

Membership Application

FRONT

Membership Application

BACK

Volume 22 ● Number 4
December 2001

FEATURES:

NEWS:

REGULARS: