

news@UK

The Newsletter of UKUUG, the UK's Unix and Open Systems Users Group

Published electronically at <http://www.ukuug.org/newsletter/>

Volume 13, Number 2

ISSN 0965-9412

June 2004

Contents

UKUUG Secretariat	2
New Council Chairman	2
UKUUG Lobbyist	2
UKUUG – an update	3
UKUUG Open Source Award 2004	3
EGS and Open Source Development	4
SANE 2004: announcement	5
2004 USENIX Annual Technical Conference	5
UKUUG Tutorial: Advanced topics in DNS Administration	6
Book review: Network Security Assessment	6
Book review: qmail	8
Book review: Secure Coding: Principles and Practices	8
Book review: Cascading Style Sheets: The Definitive Guide	9
Book review: Squid: The Definitive Guide	10
Book review: WebLogic: The Definitive Guide	11
Book review: Mac OS X Panther for Unix Geeks	12
Book review: The Perl CD Bookshelf	13
Book review: SQL Pocket Guide	14
Book review: Linux Pocket Guide	15
Book review: Hardware Hacking Projects for Geeks	15
Contacts	18

UKUUG Secretariat

Jane Morrison

It has been another busy time for UKUUG. Since March we have been concentrating on bringing the details together for the Linux 2004 Conference (Leeds 5th - 8th August), the full information booklet and booking has been sent to all members and you can also find all the up- to-date details on our web site.

<http://www.ukuug.org/events/linux2004/>

Delegate bookings are arriving each day. Don't forget, if you wish to take advantage of the early-bird Tutorial and Conference fees you must book by 30th June. The Clarence Dock Bed and Breakfast option for just £ 17 + VAT must also be booked as soon as possible to guarantee this low price.

In April we organised another DNS tutorial. Jim Reid, tutor, has provided tutorials for the UKUUG members on quite a few occasions and as usual we had a good number of attendees who found the day very interesting and worth while.

The next event for your diaries is the UKUUG Annual General Meeting which will be held this year on Thursday 23rd September at UCL London at 6.00 p.m. Further details, Agendas etc. will be sent to you automatically.

Tina Bird will be providing a tutorial 'Building an Enterprise Logging Infrastructure' for UKUUG members on 14th October in London. Please see the information leaflet enclosed with this Newsletter. The next Winter Conference is planned for February 2005; we are currently looking at venues in Birmingham.

If you wish to have something published in the next issue (September) please note the copy date is the 3rd September.

New Council Chairman

Charles Curran

It is my great pleasure to let you know that at UKUUG's Council meeting in April I successfully managed to resign as Council chairman, a post Council had (re-) elected me to since 1999. I will remain on Council until my term of office expires at the AGM in September.

The new chairman is Ray Miller, who works at the University of Oxford's Computing Services. Ray has been doing really good stuff for UKUUG, such as putting together the recent, very successful winter conference.

Apart from his IT interests, he is known to drink real ale and ride a bicycle, although not at the same time.

I am sure that you all wish him well, and will help him in whatever way you can.

UKUUG Lobbyist

Charles Curran

You will probably have noticed that we have been discussing whether UKUUG could profit from having a lobbyist or campaign officer, particularly to work on the free software and open source front. Many thanks to those of you who have written recently to share your opinions with us. There is some doubt that this activity would really profit UKUUG and its membership

especially as it could be relatively costly. However, Council has decided to try it out over the next few months. Initially, at least, we are asking Leslie Fletcher (Manchester) to start work on this. We hope to have more to report in the next issue.

UKUUG – an update

Charles Curran

Those of you reading UKUUG's web pages and other 'About UKUUG' statements may have noticed that we have updated our explanation of UKUUG and what it stands for. Out has gone 'UK Unix Users Group', since although the group may have had its origins there, it is a much broader grouping, and even the group's legal name since the early 90s has been just UKUUG (Ltd).

The Memorandum of Association lists as UKUUG's goals:

- to promote and advance the knowledge, use and application of open computer operating systems using compatibility techniques pioneered by the Unix operating system
- to facilitate the exchange of information and views on the use and development of open systems
- to inform public opinion upon the subject of open systems
- to provide a focus for the standardisation of techniques used in open systems
- to encourage internationalisation of open systems for the benefit of users of open systems

So, we have concentrated on the open systems aspects. The brief expansion we give for UKUUG when we bother is "the UK's Unix and Open Systems User Group". Where we use a longer explanation, we use:

"UKUUG - the UK's Unix and Open Systems User Group - is a non-profit organization and technical forum for the advocacy of open systems, particularly Unix and Unix-like operating systems, the promotion of free and open-source software, and the advancement of open programming standards and networking protocols."

See also:

<http://www.ukuug.org/about/>

UKUUG Open Source Award 2004

Charles Curran

The UKUUG 2004 Open Source award for a significant contribution to free and open source software has just been made to Julian Field of the University of Southampton for his work in creating, developing, and supporting MailScanner, the highly respected e-mail security system, and about which Julian gave a paper at the last Winter conference. The judges noted as "Highly Commended" the Enterprise Groupware System developed by Jake Stride (Warwick University) while a student at Newcastle University. Julian wins a GBP 500 cash prize, and, thanks to support from O'Reilly, organisers of the convention and Gold Sponsors of the Open Source Award, a trip to this summer's Open Source Convention in Portland, OR, USA. Jake wins a GBP 100 book prize, also donated by O'Reilly, and a £ 100 cash prize.

MailScanner is a complete e-mail security system designed for use on UNIX/Linux e-mail gateways. It protects e-mail clients against viruses and can detect almost all spam. With e-mail

viruses costing businesses millions of pounds every year and spam accounting for around 60% of all e-mail traffic, MailScanner is the front-line of defence at more than 20,000 sites.

MailScanner has been in continuous development for almost four years. In that time it has grown from a simple virus scanner with 1200 lines of code to a complete email security and anti-spam system of over 30,000 lines. It supports the use of any combination of 20 different anti-virus engines to give the best possible coverage - commercial e-mail systems rely on one or two. It incorporates SpamAssassin, widely regarded as the best anti-spam engine available, and over 800 heuristic spam-detection rules.

Robustness and reliability are of great importance in any software system that handles e-mail, where legitimate content is often transient and business-critical. If an email message is destroyed in transmission, vital information can vanish without anyone noticing. Strenuous efforts have been made in designing and developing MailScanner to ensure that there is no chance of e-mail messages being lost due to failure of any part of the software.

MailScanner has been deployed in over 60 countries, and is used for scanning mail destined for all seven continents (even Antarctica). It scans over five billion messages per week for numerous government departments, corporations, non-profit organisations and educational institutions. It is used by large ISPs and mobile telephone companies in the UK and Europe, along with the largest space agency. It is now downloaded over 20,000 times each month, a total of more than 250,000 downloads.

MailScanner's home page is at:

<http://www.mailscanner.info/>

More details of the Open Source Award are at:

<http://www.ukuug.org/osa/>

EGS and Open Source Development

Jake Stride

As part of my undergraduate degree at The University of Newcastle upon Tyne, I undertook a final year project studying Open Source Development techniques. In order to study developer and user interactions, a software system was needed to develop a community in which to study the techniques used in open source development.

Due to the restrictions placed upon the project by university regulations, a software system had to be developed from scratch (instead of taking over the development of an existing project which the maintainer no longer had the time for). For this reason Enterprise Groupware System (EGS) [1] was developed.

At a web-development company in 2001, one of the projects on which I worked was developing an online, project-management system. With only a couple of developers working on it, it was a wonder the software ever made it out of the door; thereafter, it was continually being fixed for major bugs. With this in mind, and using the experiences gained, I decided to develop a similar system using open source development techniques.

There are several web-based project-management systems already available to the open source community, so instead of re-inventing the wheel, it was decided to combine project management tools and Customer Relationship Management (CRM) functionality to fill a gap in the software offerings. The result was EGS, which up to June 2004 had been downloaded by almost 4,500 people and the website viewed over 166,000 times.

With a software system in place and being actively developed, it was then possible to study the interactions between developers and users, and to produce a study of the benefits and poten-

tial pit-falls of using open source development techniques to build software in a commercial environment.

The major problem of developing software with strict deadlines using open source techniques is getting developer contributions built and tested on time. For this reason, the study suggests that commercial outfits intending to use open source techniques develop the core of the system in-house, whilst allowing external developers to develop modules and software for the system that are not bound by the deadlines. This way, the software still benefits from the contributions of the external developers and the bug fixing skills they can offer whilst allowing the software to be developed on time; it has the added bonus of getting additional functionality for free from the open source community.

<http://egs.sourceforge.net>

SANE 2004: announcement

UKUUG Administration

We have received the following announcement from Walter Belgers (Program Chair 4th International SANE Conference)

Coming September, the bi-annual International System Administration and Network Engineering (SANE) Conference will be held for the fourth time. The location this time is the RAI Centre in Amsterdam, the Netherlands.

September 27th will be the first of three days filled with in-depth tutorials in several parallel tracks. Following the tutorials are two days of technical conference, with high quality invited talks and refereed papers. These will be presented in two parallel tracks. The technical conference starts off with a keynote address by Paul Kilmartin of eBay.

During the tutorials and conferences, there is ample time to organise your own Birds of a Feather (BoF) or The Guru Is In sessions to catch up with colleagues in your field. You can also catch up with new technologies as explained by those people who have sent in a poster for the poster sessions.

On Wednesday September 29th, a Free Software Bazaar will be held, which is free for anyone to visit. Richard M. Stallman of the Free Software Foundation will give a lecture, and several Open Source and Free Software groups will be present.

SANE'2004 promises to have a very interesting line-up of speakers and topics and offers a great opportunity to meet and talk with well-known people from the field as well as colleagues from all around the world. For more information, please see our website at

<http://www.sane.nl/>

2004 USENIX Annual Technical Conference

The USENIX 2004 Annual Technical Conference takes place between June 27th and July 2nd 2004 in Boston, MA, USA.

Full details are available at

<http://www.usenix.org/usenix04/>

UKUUG Tutorial: Advanced topics in DNS Administration

Alain Williams

Tutor: Jim Reid. Location: London, 20 April 2004

I am self-taught in the use of Bind, gradually having learnt new bits as I needed them. What I wanted from this course was to be given general advice on the common errors or omissions that people like me make and also to learn about secure DNS since that might look good on my CV. Overall my objectives were met.

Jim taught about Bind 9.3, which at the time was still in beta. He explained that there was no excuse to not have an up to date version and digressed into one of several rants: this one about Unix vendors that still ship old versions with known exploitable security holes. Jim also recommended use of one other DNS server for the sake of diversity.

None of the about 30 delegates had a problem with the prerequisite of knowing DNS basics, which meant that we did not waste time on basic questions.

Jim started with explaining the logging subsystem. Generally, this works out of the box, but I now know how to stop my system log clogging up with messages about 'lame servers'. Then followed a discussion about rndc (the name server control program), this is something that I had never really bothered with, just doing a reload with the init script whenever I changed something. The nice thing that I learnt was how to reload one particular zone if I *know* that it has changed and do not want to wait for it to timeout.

In one of several diversions, Jim was adamant that the use of forwarders was a bad idea. I had always assumed it a good idea to use the cache in your ISP's DNS servers. Jim explained that it caused several problems – these I had noticed and I have not seen since the evening of the course.

Jim then talked about how to handle the common case of having to serve up different results to a domain depending on where the query is coming from; e.g. you want to hide your internal DNS from the outside world. He showed several ways ending with 'views'; I gave myself a pat on the back – I had got that right. We then moved on to setting up a root server, bind server security, Access Control Lists, Dynamic DNS, the lightweight resolver, transaction signatures.

The final session was an explanation of DNSSEC. Basically, the purpose is to prevent spoofing of DNS information: think what you could do if you made a bank's DNS entry point to your servers. The proposed solution seems close to what will be finally implemented, it is basically a mechanism for signing DNS records with an authentication chain back to the root servers in a way similar to an encrypted web page. This makes getting and verifying DNS entries more expensive but this should be largely mitigated due to caching. I understood what Jim said at the time, but it is something that I must actually do before I can really make that claim. I do believe that this is a path that we must tread as we become more reliant on the Internet in an increasingly hostile world.

The venue was a pleasant suite in a central London hotel with a decent lunch.

Network Security Assessment

Chris McNab

O'Reilly and Associates

ISBN 0-596-00611-X

396 pp.

£ 28.50

reviewed by Andrew Cormack

It has often, and probably accurately, been suggested that intruders looking at networks from the

outside know more about them than the managers whose job it is to run them. This new book may go some way to redressing the balance. Chris McNab explains how network managers can use the same techniques and tools as the intruders to examine their own networks and identify security weaknesses. With any tool having both good and bad uses it is important to be clear about the differences. Here it is best stated in the preface: intruders use tools to make security worse, network managers use tools to identify problems, fix them and improve the process that failed to prevent them being there in the first place.

The book begins with the process of host and network enumeration: simply finding out what networks and computers there are out there. Much of this information can be obtained from public sources such as whois and DNS databases without even visiting the target network. Even Google can be used for information gathering, though much of what is discovered is likely to be useful for social engineering attacks, which are not the main focus of the book. Traditional port scanning can then be used to confirm which hosts and services are active: the author gives a good review of the various techniques, their speed, accuracy and visibility.

Most security vulnerabilities arise in network services, so the bulk of the text concentrates on these. Grouping services into chapters by function is an excellent idea, which allows the principles as well as specific problems to be covered. Details of particular attacks will go out of date, but it is surprising (and depressing) how many old bugs are still out there on the Internet. Chapters describe remote information services, web, remote maintenance, FTP and databases, windows networking, e-mail, VPN and RPC services. In each case information can be gathered using standard system tools, auditing software or exploit programs: most of these can be downloaded from the Internet. For those who want to know how exploits work, there is a detailed technical description of buffer overflows, integer overflows and format string bugs, which is sometimes hard going but worth persevering with. Each chapter ends with a useful checklist of countermeasures that system and network managers should be using to protect themselves against the attacks described.

The final chapter brings all this together by walking through the security assessment of a small network, identifying weaknesses and recommending short and long term security improvements. The approach demonstrated, first identifying active machines, then identifying the type and version of each of the services they run, provides a sound framework for anyone performing their own assessment. Every assessment should have a clear objective, defined in advance, and this will control what techniques are used. If the objective is to ensure that a firewall is correctly configured, for example, it will not be necessary or appropriate to research or exploit individual vulnerabilities in services accidentally left exposed. The truly expert security analyst knows when to stop.

The book mentions some very powerful tools, and occasionally blurs the line between proper and improper use. To place it firmly on the side of the good guys there should be a mention of which tools can be safely used on a production network, as aggressive scanning or probing of vulnerabilities can cause systems or routers to crash. A reminder that many of the techniques described are illegal if not done with proper authorisation would also be welcome. Many readers of the book will be horrified to learn just how leaky their computers are: if nothing else the book is an excellent argument for a robust firewall. Others will want use the tools and techniques to learn how to make their own networks secure. Before you do this, please make sure you are entitled to do so and be very careful not to mistype your IP address and probe the network next door!

qmail**John Levine****O'Reilly and Associates****ISBN 1-56592-628-5****248 pp.****£ 24.95****reviewed by Lindsay Marshall**

Deciding on an MTA is, of course, a deeply serious and important, religious matter. There are those who swear by Sendmail, the one true faith, but more common are those who swear at Sendmail and move to Postfix, Exim or qmail. Even more common are people who no interest at all in the MTA that they have to use and leave that kind of thing to other people. If you are one of the latter then this book will be of no interest to at all, so you can stop reading this review now. Indeed, this is also true for devotees of Sendmail, Postfix and Exim who only agree on their hatred of qmail and will have no truck with heresy. However, if you have looked at all the MTAs and have decided to go with qmail, then this book might well be of interest to you. I only say might because there doesn't seem to be much in the book that you can't find on the net. Nevertheless, the print copy does have the advantage that everything is one place (Doh! obviously Ed.), there are a few more examples, and there is a stylistic homogeneity which makes the reading easier than trying to get to grips with different webpages.

The big question is do you actually need this book? Many people who run qmail do so because it is extremely easy to setup, is pretty secure, and once you have it running you may never need to touch it again if your surrounding environment doesn't change drastically. (I don't want to get into the issues about why you may not want to run qmail). I run qmail on three machines which don't need complex mail setups, their installations are essentially clones of each other and I've not had to change anything for years. Recently I wanted to add a new mail feature to one of my systems (don't ask) and had a small amount of trouble working out how to achieve what I wanted. I eventually found the answer via the net and trial and error: looking at this book, I don't think that it would have helped me, so I am even less convinced of its necessity.

As always with O'Reilly the book is well produced <Insert boilerplate O'Reilly review text here> and the technical content is accurate, but I cannot in all honesty say that you need to have it if you want to install and run qmail : it will sit on your shelf unread until you move office. For completists only.

Secure Coding: Principles and Practices**Mark G Graff and Kenneth R van Wyk****O'Reilly and Associates****ISBN 0-596-00242-4****224 pp.****£ 20.95****reviewed by Ray Miller**

This slim volume contains a wealth of information that will be of interest not only to software developers, but to anyone responsible for the deployment and operation of computer systems. Indeed, if I had one complaint about the book it would be that the title is misleading: it is not so much about secure coding as the overall software development process, and relatively little is said about coding per se. But do not let this put you off, programmers will benefit from reading this book too.

The first chapter gives an overview of security vulnerabilities, covering different types of attack; how vulnerabilities arise; who might want to exploit them; and how we can defend our systems. Further chapters cover different aspects of the software development process: architecture, design, implementation, and testing. Every chapter contains useful information and practical advice, and the authors draw on their extensive experience to back this up with examples and case studies.

They advocate a holistic approach to security: you cannot work around design errors at implementation time, and the best design and implementation can be laid waste by poor operations procedures. This theme permeates the book and is reinforced by the chapter on operations, which emphasises the importance of providing a secure environment for running a business application.

Most chapters also include sections on good practices and bad practices. I found that these conveyed information effectively, particularly where they drew attention to failings in my own organization.

The book is concise, with clear explanations of new concepts. Ideas are further clarified through the case studies, which add real-world interest to the book. To top it all, an appendix extending to 10 pages lists further resources.

In the prefix, the authors remark that they want the book “...to be `_read_`”. They have certainly succeeded in producing a very readable volume, and I encourage my fellow software developers and systems administrators to go out and read it. As for myself, I shall be making time to follow up the resources - and putting some of these principles into practice on my own systems.

About the reviewer: Ray works as a Unix Systems Programmer at Oxford University Computing Services where he leads the Systems Development and Support team. His responsibilities include development and day-to-day running of the University's central mail store, web servers, and interactive GNU/Linux service. He is a strong advocate of Free Software, particularly the GNU/Linux operating system.

Cascading Style Sheets: The Definitive Guide

Eric Meyer

O'Reilly and Associates

ISBN 0-596-00525-3

528 pp.

£ 28.50

reviewed by Mike Smith

It is worth having a look at the CSS Zen Garden website – especially the “Wiggles the Wonderworm” design !

<http://www.csszengarden.com/>

This demonstrates how the layout of a page can be fundamentally changed using CSS. It's much more than just changing font faces and sizes. NTK had a play around with CSS a while ago, but that's about as far as I thought you could go.

This is a second edition book - unfortunately I don't have the first edition so cannot compare it. However, apparently it's not much bigger, and some support sections were dropped from the first edition (that the author considers better online anyway, so that it can be kept up-to-date) - enabling a lot of additional information to be added to the book. CSS2 and CSS2.1 are covered. There's been a gap of four years between these editions - it's surprising how the time flies.

The idea behind CSS is that HTML can go back to its original conception – describing the structure of a document, not its presentation. CSS lets you do anything you could do with FONT and all the other tags, but it'll do much more too.

CSS is great for content management systems. I use PostNuke (which was hacked twice recently, but that's another story), and it's easy to set colours, style etc in one place, and it's reflected across the whole site. You do get a similar effect with HTML templates, of course - but it can be easier with CSS. What's quite interesting is that you could refer to a CSS in an XML document and do away with HTML altogether.

At the risk of telling you something you can easily find on the O'Reilly website, the book has chapters covering some generalities, then specifics on Fonts; Text; Elements; Borders; Colours (without the 'u', unfortunately); Positioning; Tables and Lists. There are some interesting chapters at the end on changing cursors and other system things, and support for other media - such as Audio (though apparently specific Aural support is being replaced by something more generic). Although the obvious use of audio is to aid the visually impaired, you can even position the source of sounds in 3D space - so you could have some fun with 3D sound systems. I've just got a Creative Audigy 2 ZS, and it is fantastic with first person shooters - though perhaps web browsing isn't going to be quite that exciting.

There are lots of examples (albeit in Black and White), and the text serves two purposes - you can read through it initially to learn about Cascading Style Sheets, and also use it as a reference manual.

Although 500 pages seems like a lot, it is broken down well, and there's also an appendix with a complete reference of CSS properties.

I do recommend this book. A lot of the information will be available on the web of course - but if you like paper, this is a good one.

Squid: The Definitive Guide

Duane Wessels

O'Reilly and Associates

ISBN 0-596-00162-2

464 pp.

£ 31.95

reviewed by Mike Smith

This is a very good book, covering the basics of Squid all the way through to some fairly advanced topics. It is in a pretty standard format – similar to other O'Reilly texts on software packages. The first three chapters provide some context and how to obtain, compile and install Squid. I hadn't realised that NetApp's NetCache has the same code base, originating from the Harvest project. Each chapter has a few exercises at the end. I presume Duane is still in education and intends for each of his students to buy a copy - a common trick!

Naturally, on the cover there is a picture of ... a squid! Apparently they can grow to 60 feet in length, weigh up to a ton, and their eyes can be as much as 10 inches in diameter. The Colophon is always worth a read.

Anyway, chapter four is a good inclusion in the book - a getting started quick section. i.e. what the essential configuration parameters are to get you going. The full list of parameters is eventually covered over a number of chapters, organised into a number of subjects (such as access control, disk caching etc) - thankfully the book isn't just an alphabetical list of these parameters and an associated explanation. In fact appendix A is a reference for all configuration parameters.

There's a section on how to setup interception (transparent) caching. Although this is, in the main, not about Squid itself, obviously you need to know how to setup IPChains/IPTables and other mechanisms (for instance configurations for Alteon, Extreme, Cisco and other devices are briefly documented) to support a caching service. Its worthwhile having, but because it tries to cover so many routing appliances, it's a little scant on detail.

Things get more interesting when there are multiple caching servers involved. This isn't something I've done myself - I've only used a simple single-server configuration to avoid using work's official Internet gateway in the past ;-). The mechanics of setting up ICP (and other protocols) are covered, but I would have welcomed a more thorough discussion on the "macro" architecture of caching services - by that I mean the overall topology. For instance, in a large network environment with multiple external gateways in different locations, would it be best to distribute the caching service, with servers near to the firewalls (in network terms) or have a single centralised farm? Please write in to the magazine if you know - but I suspect the answer is ymmv. There could have been more diagrams in this area, for instance to describe various hierarchical, peering and load-balancing options too.

There's a brief chapter on Squid Redirectors, which looks like an interesting feature. Never used it. There is, again a brief, chapter on Authentication. This just covers the Squid aspects rather than a full explanation of an entire implementation. So just that you need to configure and compile in the helper programs, and use various configuration file options to use them.

Another tool that I haven't used, but looks extremely useful is the Cache Manager. This has web and command line interfaces for monitoring Squid's performance. Statistics are maintained for all manner of metrics including DNS resolution times, disk IO, memory and network utilisation. I'm not sure how well it would tie together the statistics from a cache farm or hierarchy - but I suspect that's another exercise for the interested reader.

I'm most interested in the datacentre environment, and chapter 15 covers one area that can be really useful - Server Acceleration. There are some specific configuration parameters to deal with this, and some restrictions. Good to see it documented.

So in summary, as I opened the review with, it's a good book and recommended, but lacking in some areas - particularly in the design of caches with multiple caching servers. Maybe that's in Duane's other O'Reilly book, "Web Caching" - I really ought to review that!

WebLogic: The Definitive Guide

Jon Mountjoy and Avinash Chugh

O'Reilly and Associates

ISBN 0-596-00432-X

848 pp.

£ 35.50

reviewed by Mike Smith

WebLogic is a very sophisticated J2EE platform. I'm not sure how it compares with the Open Source JBoss, but it has some advanced clustering, security and management features, and all the usual J2EE services. This book tries to cover it all, hence the 800 or so pages.

I don't think I'm giving away any secrets when I say that the UK Government's DotP (Delivering on the Promise) web services are based on WebLogic. (UKOnline, one example of these services, used to use Vignette, but not any more. You can tell when it's Vignette because of all of those commas in the URLs, btw.)

The UKUUG has different groups of members - SysAdmins, Developers, and I guess Architects. This book provides information for each of these groups. It does expect some prior understanding of application servers - it is not a basic guide, and to some extent it leaves some information

out and is therefore incomplete (and not “Definitive”). For instance there are no detailed installation instructions – only guidelines. I’ll accept this, as the book will complement the WebLogic documentation, of course.

There are some code and XML fragments, and a few diagrams. However the vast majority of the book is presented as a title and a descriptive paragraph (or several paragraphs) about the subject matter at hand.

Although I tried, I just couldn’t get into this book. I found it quite dry and boring. Possibly because it is heavy reading, but also because it covers the three target audiences outlined above. Therefore probably only a third of it is of direct interest to the reader. (For example there is a lot of J2EE jargon, and similarly J2EE concepts that a non-developer, like myself, doesn’t want to know about!)

Even the potentially most exciting subjects (for an infrastructure person), such as setting up for clustering and load balancing, don’t seem to be covered well. WebLogic really shines here because you can cluster EJBs across different servers and WebLogic keeps track of everything for you. There are considerations when you’re using, say, Apache to load balance across multiple servers (there’s a WebLogic plugin you use), so there are a lot of options and considerations to be covered here.

I can’t therefore recommend this book to you. The fact that this review is relatively short demonstrates that I couldn’t find the interesting and exciting bits anywhere; and that normal O’Reilly sparkle just isn’t there.

Mac OS X Panther for Unix Geeks

Brian Jepson and Ernest E Rothman

O’Reilly and Associates

ISBN 0-596-00607-1

384 pp.

£ 17.50

reviewed by Jan Wysocki

Well I’m not sure that I’m very happy to think of myself as a geek, but as Unix SA who’s recently acquired a dual G4 Mac I thought this book would help me get more out of it. I bought the G4 largely to replace two ageing Macs, but also with the thought that I could de-clutter my house and maybe get rid of some Linux boxes as well.

Who’s this book really aimed at? Well I think it’s fine for someone like me who’d just like to integrate some GNU clients like the Gimp and the Gnumeric spreadsheet, but it should suit someone who wants to do some serious coding for this platform. Which also means that as it’s only a 300 page book its aims are probably higher than its achievement. With information aimed at several audiences, there’s going to be some dissatisfaction.

Let’s start with what the book has done for me in the few days I’ve had it. As it happened I was about to make my first attempt at a compile and install on OS X, so I checked out the chapter on building applications, then noticed that Fink was dealt with a little later. I’d heard of it, not understood what it was except that it was used in a porting context. A quick read put me straight, I wasn’t surprised to learn that it’s a package manager, but hadn’t realised that it would lead me to a raft of compiled packages at Sourceforge. I quickly had Fink installed, rapidly followed by a binary installation of Gnumeric. The Fink chapter gave me enough information to understand and use Fink, leaving it to the Fink web site to fill in the details.

I haven’t read this book from cover to cover. It works well as a source of specific information when you know enough to know what you want to do but need platform specifics. As far as I can

see that's how you'd use this book. It does follow a plan being divided into parts with different aims and there's a lot to be said for at least reading Part 1: "Getting Around" which introduces Mac OS X from a user's perspective. Like any book, its indexing isn't comprehensive enough. After learning about the X11 preferences menu, I found it easier to discover how to stop the X server from starting in 'rooted' (full screen) mode by using 'find' to identify the relevant config file, than to find the page that held the information I needed.

I thought that the chapter on System Management could do with additional information, but that the chapter on Directory services was spot on. A chapter on filesystems would be useful. There's an appendix but that just lists principal directories whereas I'd have liked clarification on topics like forks, fsck caveats and how /Volume works.

In places the information can be quite dense. Superficially some of the appendices seem like fillers, but there are gems in there, waiting to be mined. So without scanning Appendix 3 I might never have guessed that 'pbcopy' and 'pbpaste' are available as tools to access the 'clipboard'

I received this book just before becoming aware that a new OS X release was looming. This book does contain references to differences from the previous release of OS X, e.g. how to avoid problems with CPAN, and a publication date about 6 months after Panther, which suggests that considerable checking went on, as the book was revised.

If you're a Unix user who needs to go beyond the scope of "The Missing Manual" then this is probably the book for you. Developers new to Mac OS X will find it a good introduction before consulting on-line material.

Jan Wysocki has been administering Unix systems since the days of A/UX.

The Perl CD Bookshelf

Various

O'Reilly and Associates

ISBN 0-596-00622-5

600 pp.

£ 71.50 plus VAT

reviewed by Mick Farmer

For general text processing and simple number crunching I still use Perl, and the earlier edition (circa 1998?) of this CD sits in one of my CD drives for weeks at a time. Therefore, I was keen to review this latest version to see what has changed in the Perl universe over the last few years.

The six books on the CD are listed below and the Bonus (real) book is Perl in a Nutshell.

Perl in a Nutshell, 2nd Edition

Learning Perl, 3rd Edition

Programming Perl, 3rd Edition

Perl Cookbook, 2nd Edition

Mastering Regular Expressions, 2nd Edition

Learning Perl Objects, References, and Modules

Each of these books has already been reviewed in the newsletter, so I will concentrate on the complete package.

What immediately springs to mind is that the books are completely platform-independent. The last edition included "Learning Perl on Win32 Systems" and I think this demonstrates how Perl has matured with virtually all low-level operations now being done "the Perl way", rather than relying on the underlying OS to trap the unwary porter. For those with long memories,

this process has been continuing for some time – I can still remember the “Perl Resource Kit (UNIX Edition)” which contained five (real) books and a CD of software from Larry Wall.

The other book that has been dropped is Sriram Srinivasan’s “Advanced Perl Programming”, the book that introduced me to Perl’s object-oriented view of the world and to Perl’s internal structures. [We understand that a completely new edition has been written – Ed.] This has been superseded by last year’s book from Randal Schwartz and Tom Phoenix entitled “Learning Perl Objects, References, and Modules”, a thoroughly up-to-date introduction to this field.

The sixth book in the collection is the updated “Mastering Regular Expressions” by Jeffrey Friedl (in PDF format). This has expanded sections on Perl, Java, and .NET, but I’m not sure what it offers to the Perl programmer. If I need help with a Perl regular expression, I click on (it used to be reached for) “Programming Perl”, which is my Perl Bible.

All in all, a good update to a trusted friend, but have you seen the price?

SQL Pocket Guide

Jonathan Gennick

O’Reilly and Associates

ISBN 0-596-00512-1

160 pp.

£ 6.95

reviewed by Lindsay Marshall

I must own up to being so old that they hadn’t invented stuff like SQL when I was still being taught things so to do anything with databases I rely entirely on information in books. (I have learned from experience that asking people about things like complicated SQL joins just makes everyone’s heads hurt.) Since I tend to do simple things (largely because I don’t understand the complicated things of course) I need simple books. Ones that tell me what happens but not too much detail about why: mostly, I really don’t care why so long as I get the data I want in a reasonable time, that’s the beauty of databases. I also use MySQL, though I believe that other products are available (Oracle, DB2 and SQL Server in this book).

So when I want information, I sometimes go to the MySQL documentation: online, free, sometimes has useful comments attached but mostly way too terse and has no explanation at all. If that fails, I turn to SQL In a Nutshell, which is OK but often fails to tell me what I need to know. From now on, however, I shall be turning to this book first. It is almost “just the facts, ma’am” but with a light dusting of examples and explanation. Loads of Oracle stuff (reflecting the size of the product more than its market position) info of course (as well as other two), but the MySQL part is big enough.

The book is nicely sized, like all the pocket guide series, and I keep picking it up and flicking through, and, for the moment, every time I do, my eye catches an entry and I learn something. If you work with SQL but are not entirely database centred then this is the book for you. The author’s intention is for the book to be written for programmers and he succeeds. There are lots of PHP/ASP/Perl/JSP web programmers out there who would find this a useful addition to their library.

Mind you, I still don’t know why some of big joins don’t work.

Linux Pocket Guide

Daniel J Barrett

O'Reilly and Associates

ISBN 0-596-00628-4

200 pp.

£ 6.95

reviewed by Roger Whittaker

This book could be described as a “little brother” of O'Reilly's “Linux in a Nutshell”. Both books consist mainly of a summary of commands, with lists of options and example usage. In my experience, the main use for books of this type is not actually to use them to look anything up while you are working (man pages and Google are quicker than walking across the room to the bookshelf, or even opening the book to the right page). Rather, by browsing books like this in odd moments, one can be informed or reminded of commands one never knew or had forgotten existed, or can pick up hints about how to do something in a better way.

There is a temptation when writing such a book to list every option to every command. Clearly there is a trade-off between completeness and readability, but personally I would always prefer to see more examples of usage and less reprinting of information direct from the man page. However, readers who want to use the book as a desktop reference will probably hold the opposite view.

While “Linux in a Nutshell” makes a point of being generic, this book has the words ‘Covers Fedora Linux’ on its front cover. The publisher's “blurb” on the back cover states, however:

This book is tailored to Fedora Linux – the latest spinoff of Red Hat Linux – but most of the information applies to any Linux system.

I found that statement to be accurate: in fact there are only a couple of entries in the book that are Fedora or Red Hat specific, and they are clearly noted in the text as being such. Where graphical programs are mentioned, however, they do tend (as one might expect) to be those associated with a Gnome desktop system rather than KDE. My advice to O'Reilly would be to remove that reference on the front cover: it might repel more potential buyers than it attracts, and the content is useful to everybody.

This is a straightforward, accurate and well produced command reference for the most common Linux commands, and I can recommend it, particularly to new users of Linux.

Hardware Hacking Projects for Geeks

Scott Fullam

O'Reilly and Associates

ISBN 0-596-00314-5

348 pp.

£ 20.95

reviewed by Mike Smith

This is something different – not the usual software manual ... a book on hardware projects. It sounded very interesting, but I was immediately very disappointed when I opened it up.

It rapidly became clear that this book (or at least a part of it) is largely based on a number of projects that have been collected from various places on the web. If you've been around as long as I have (and I'm sure you have) you come across these things all of the time – whether just cool sites, pointers from Slashdot, or whatever. Examples include a Macquarium, Furby Hacking and Home-made 802.11b Antennas.

Some other projects are just naff - a portable laptop power supply (i.e. put some batteries in series with a suitable connector), and a periscope for the car (ridiculous).

Having got the major issues out of the way, I'll continue to grumble as I go through the book.

There are three major sections to this book - easy projects, hard ones, and appendices. The easy section has 6 projects including the battery thing and periscope. Pah. The others are the Mac Aquarium the 802.11b Antennas, Furby and a PC Water-cooling system. Of course you can just buy a water-cooling system if you want one - far safer I would imagine. So the only ones I find of interest are Furby and wifi Antennas - both of which are repeats of information we read on the web some years ago.

There are 9 advanced projects. Building a digital video recorder is interesting, but a bit old hat. MythTV isn't mentioned either. There are some nice ideas at the end about using IR or wireless PDA for remote control, but bluetooth ought to be an option these days - then your 'phone could be used to change channel, record, pause etc.

There's a project on constructing a building-sized display - using lights in the windows of an office block (as at the end of the film Hackers). Good, but yet again a repeat of information on the web.

Cubicle Intrusion detection system: Boring (basically a magic eye with a lamp attached to it).

Internet Toaster. Great, I thought: give it an IP address, it'll run a web server, set the lightness and turn the toaster on from my phone. No - just a rudimentary selection of two patterned messages for the toast (using a mask in front of the elements).

How to build a home arcade machine. See Google. Its just Mame and some MDF (from B&Q.)

Building a wearable computer might be interesting, but makes you look a bit nerdy, so I'll avoid it for now, thank you. (In fact I still haven't got used to those bluetooth headsets that look like a beetle on your ear either.) The chapter on this just lists the many options for the machine, input and output devices, so not very good anyway.

The very last project is probably mainly there for nostalgia - building an Internet coffee maker (like the Trojan room, obviously). However it does try to go one better - instead of a webcam, probes are used to monitor coffee level and temperature. I don't think much of the tubes going into the coffee, but I did like the look of SitePlayer, which I've not come across before. (Its a mini webserver that plugs into a network and has a serial port. That's the sort of thing you really need for the Internet toaster too.) Besides, I've got a Gaggia, so don't need this - I just press a button in the morning for my coffee!

<http://www.gaggia.it>

There are two projects that I've not covered yet. These are building a Remote Object Tracker, and making remote control cars play laser tag (with Infrared, rather than with lasers). I've left them until last as they are the ones that I find the most interesting. The latter covers programming of PIC controllers, which would be good to try out one day. This project does include the construction of a radio transmitter (as a trigger for the Infrared gun) - so I don't know how feasible or legal it will be to do this in the UK. It seems a bit odd building another radio transmitter when you've already got one to remote control the car. I'm not an expert, but thought you had several channels to play with.

This just leaves the Remote Object Tracker, which uses a combination of GPS, PDA for viewing, and PMRs. I hadn't thought of using PMRs to transmit location information, so thought it quite ingenious. The range of PMRs is relatively short - a mile or two in good conditions, so it has limited uses. The interesting bits are the conversions from GPS to PMR, and PMR to PDA. The first uses a commercial kit called a TinyTrack - I wonder how much he's getting for promoting that! This has a serial interface for the GPS and phono in (and out) for the PMR. The decoding at the other end is performed by a TNC. Again there's a recommended kit. Put some mapping software on the PDA (he uses a Palm) and you're all set.

I had hoped to actually do one of the hacks in the book for the review, send pictures in for the magazine etc. But actually, the book isn't very good, and I found nothing worth the effort.

Contacts

Ray Miller
Council Chairman; Events; Newsletter
Oxford
Tel: 01865 273 200
ray.miller@ukuug.org

Charles Curran
Events; Newsletter
Oxford
Tel: 07973 231 870
Tel: +34 954 371 381
charles.curran@ukuug.org

James Youngman
UKUUG Treasurer
Manchester
james.youngman@ukuug.org

Sam Smith
Website
Manchester
sam.smith@ukuug.org

Alasdair Kergon
Events
Reading
alasdair.kergon@ukuug.org

Alain Williams
Watford
alain.williams@ukuug.org

Roger Whittaker
Schools; Newsletter
London
roger.whittaker@ukuug.org

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273 475
Fax: 01763 273 255
office@ukuug.org