



news@UK

The newsletter of FLOSS UK, the new name for the UK's oldest Open Systems User Group, UKUUG

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

Volume 20, Number 3

ISSN 0965-9412

September 2011

Contents

From the Secretariat	3
Chairman's report	3
Intensive Introduction to Google App Engine	4
NLLUG event notification	5
Open Source Monitoring Conference 2011	5
Fill your boots with RAM	5
Patent madness	6
Why am I selling Tux Ties?	7
Book review: Asterisk: The Definitive Guide	8
Book review: The Book of PF	11
Book review: Javascript: The Definitive Guide	13
Book review: Agile Web Development with Rails	14
Book review: Learning Android	15
Book review: The Book of CSS3	16
Book review: iOS4 Programming Cookbook	16
Contributors	17
Contacts	19

From the Secretariat

Jane Morrison

We continue to work with Josette Garcia at O'Reilly to organise tutorials.

Our next tutorial, 'Building the network you need with PF, the OpenBSD packet filter' will be presented by Peter Hansteen on Thursday 22nd September at the Imperial Hotel, Russell Square, London.

At the time of writing we still have places available.

The AGM this year will also be held on Thursday 22nd September at the Imperial Hotel, Russell Square, London starting at 18:15. Full details including the Agenda have already been sent out on paper and are also available on the web site.

Looking ahead please see below a list of other forthcoming events:

- Saturday 8th October – unconference – Manchester Conference Centre
- Wednesday 16th November – 'Practical DNSSEC Tutorial' tutor: Jan-Piet Mens – London
- 19th–22nd March 2012 – Spring Tutorial and Conference – Edinburgh
- 12th April 2012 – 'Intensive Introduction to Google App Engine with Python' tutor: Paul Barry – London

Full details of all the above events can be found on our web site.

The next Newsletter will be the December issue and the copy date is: 18th November. As usual any articles, letters etc. can be sent for inclusion to

newsletter@ukuug.org

Chairman's report

Paul Waring

Unconference

Our second unconference will be held on 8th October at the Manchester Conference Centre, which we have used several times for our Spring conferences. Last year's unconference in Birmingham was well attended, and we have taken into account feedback which will help us to build on that success. As before, the event will take place on a Saturday so that delegates can travel to the venue and back in one day and without having to book time off work, although some of us will be going out for a curry on the Friday night and you are most welcome to join us.

If you represent or know a local FLOSS group which would be interested in attending the conference, we have some space and tables in a central area at the venue and would be keen to have you involved.

AGM

Our AGM will be held shortly after this newsletter arrives, on 22nd September in London. Niall Mansfield is stepping down at the end of his first three year term, and has decided not to stand for re-election due to other commitments. I would like to take this opportunity to thank Niall for the work he has done on Council in the last three years, particularly on the publicity and promotion side and with the introduction of the FLOSS UK brand.

Martin Houston, who was co-opted onto Council earlier this year, will be standing for election at the AGM. At the time of writing there will be one vacant place on Council at the AGM, and I would like to encourage members to consider standing for election if you would like to get more involved and feel you could help with the running of the organisation. Please do get in touch with myself or Jane if you would like to find out more about what the role entails.

Intensive Introduction to Google App Engine

On Thursday 12th April 2012, an intensive introduction to Google App Engine will be offered by UKUUG and O'Reilly at the Imperial Hotel, Russell Square, London WC1B 5BB.

Prerequisites:

Course attendees should have prior experience with Python, Web Development (HTML, CSS, JavaScript), and RDBMS technologies.

Course Content:

- Review of the Python you need for App Engine
- Preparing your computer for App Engine development
- Building a Real World App Engine webapp
- Understanding the webapps requirements
- Working with App Engine's web application framework (webapp)
- Defining your webapps data model
- Building your webapps user interface with templates
- Tying your webapp together with code
- Deploying your webapp to the cloud
- Updating your webapp with new features
- Outgrowing webapp and scaling your app
- A lightning review of the App Engine framework landscape
- Advanced App Engine features
- Wrap-up

Tutor biography:

Paul Barry lectures at The Institute of Technology, Carlow in Ireland, where he has taught for 15 years on topics such as cloud computing, web development, network programming, scripting technologies, Linux and systems management.

Prior to joining the academic world, Paul spent a decade in industry in software development and IT management roles, both in Ireland and in Canada. Paul has an MSc in Computing. He is author (or co-author) of four books: two on Perl (for Wiley) and two on Python (for O'Reilly).

Paul's latest book, part of the popular Head First Series from O'Reilly Media, is Head First Python (published in November 2010). Paul has been writing for Linux journal since 2000 and has been a Contributing Editor with the magazine since 2005.

Find out more about Paul at his website: <http://paulbarry.itcarlow.ie>

Early bird rates

- Individual and Academic members: £210.00 inc VAT
- Corporate members: £270.00 inc VAT
- Non-members: £320.00 inc VAT

See <http://www.ukuug.org/events/Googleapp/> for the on-line booking form.

NLLUG event notification

We have received notification of the following event.

Networking: IPv6 and other network related stuff

This conference (NLUUG's Autumn Conference) will be held on 20th October.

For further details, please see <https://www.nluug.nl/index-en.html>

Open Source Monitoring Conference 2011

We have received notification (from Markus Neder) of the 6th Open Source Monitoring Conference (formerly known as the Nagios Conference), which will take place in Nürnberg, Germany on the 29th and 30th November 2011.

This two day event will offer two tracks throughout the two days, and is organised by NETWAYS.

It will be held at the Holiday Inn, Nürnberg.

For further information, please see: <http://www.netways.de/en/osmc/y2011/>.

Fill your boots with RAM

Martin Houston

A few weeks ago I was tempted to buy a new PC. Ebuyer had an offer I could not resist - a 'No OS' PC with the latest 'SandyBridge' chipset, dual core Pentium 1TB hard disk and no less than 6GB of memory all for just £250.

I am pleased with the machine, the CPU is to an i3 as and i5 is to and i7, just missing the hyper-threading aspect. On raw benchmarks it is about half the speed of the top of the range i7 chips that on their own cost 2 to 3 times what this whole machine did!

I soon had the system running 64 bit Ubuntu and several VirtualBox VMs without breaking a sweat.

Opening the case up though I realised that this machine was at a bargain entry level price was just that, the new entry level. The main-board was a well reviewed but a mainstream model, no overclocking options for example and with only two ram slots, making the 6GB of memory consist of just two 3GB DDR3 SIMMs.

This caused me to look into what memory costs nowadays. Reasonable generic DDR3 memory costs less than £20 for a 4GB SIMM if you shop around.

This makes the common sense standard memory for even entry level systems with 2 slots 8GB, and the slightly more expensive 4 slot motherboards capable of 16GB of memory. Moving from 2 or 4 to 8 or 16 makes running multiple guest operating systems under VirtualBox or Vmware far more realistic.

There is also a far simpler use for extra memory that is often overlooked. Linux has a built in ramdrive called `/dev/shm`. The default configuration is to set this at half the size of the memory but it can be altered upwards with a remount command.

Being main memory `/dev/shm` is blisteringly fast, much faster even than the current hardware craze for SSD drives.

A good example of this was that I created a simple SQLite database of filesystem data to answer questions like “How many files have not been modified in 5 years and who owns them?” a full description would need another article! The main feature of the schema was lots of indexing. This made inserting data very slow, but finding it in complex ways fast. However build the database under `/dev/shm` and it is ready in about a thousandth of the time.

Once all the hard indexing work is done the database can be copied to normal disk, that same memory then used as conventional disk buffers for the read cache. Why is creating the database so slow when there is a lot of memory? The reason is that write cache or ‘dirty’ buffers can only be allowed for a few seconds. To do otherwise would risk a power outage or OS crash be in danger of causing huge filesystem corruption.

By using `/dev/shm` you are giving the OS permission for changed data to just stay in RAM as long as you choose. If the computer crashes the contents of `/dev/shm` are gone, but if you can regenerate it there is no problem, the only cost of a crash is lost time. Not bad for a thousandfold possible speed improvement!

If you have a lot of memory then using it for this sort of thing becomes possible. That £80 could pay for itself in days.

If you can increase the memory on your Linux and other 64 bit OSes then now is the time to do so.

Have lower capacity RAM left over? Why not donate it to a local charity? Someone else may find a use for it.

Patent madness

Martin Houston

Over the past few months we have seen the sort of strategies that anyone familiar with Eric Raymond’s ‘Halloween Documents’ turned not against Linux itself but as its very successful offshoot, Android.

Android has been taking the mobile phone and tablet market by storm. Exact numbers depend on who you ask and will be out of date by the time you read this – just Google for it. The smart-phone market used to consist mostly of Blackberries and iPhones but Android has now come to be market leader.

The secret of its success is of course openness. Unlike the tight manufacturer control exerted over the other platforms Android is a slimmed down Linux system with a Java based UI on top of it. Developing for that UI is open to anyone with the skills to do so using a SDK and development tools that free to download for Linux.

Anyone who can afford a PC with 3 to 4GB of memory can set themselves up as an Android software house. What’s more Google release the whole source code of each version of Android as soon as the next new one is out. You can get the full source code of Android 2.3 ‘Gingerbread’ now and ‘Honeycomb’ will be available for public scrutiny as soon as ‘Ice Cream Sandwich’ is released. It is a little bit un-free from a purist point of view but they do it to give the Android ‘shiny new’ API some stability so hardware with sexy new features that people want to buy can be developed.

Android is much more free than any of the mainstream alternatives; which is why Microsoft and Apple in particular want to see it dead!

Both have been attempting with varying degrees of success to shake down mobile phone companies like HTC for substantial per handset royalties for various patents that Android is claimed to ‘infringe’. This tactic is straight out of the Halloween Documents playbook.

Relatively new and small Mobile manufacturers seemed to have little choice but to give into this extortion and pay up. The hope of Apple and Microsoft is of course that the cumulative licensing fees would lead to the crazy situation that a ‘free’ choice ends up more expensive than the non free one.

That would be hard to explain to the buying public and many would just take the saving in return for some freedom, sadly.

This whole situation was beginning to look rather bleak with the only solution looking like political pressure to tear down the corrupt US IP system and those round the world that have been modelled on it. However on the 15th August, Google itself entered the fray. It started the process of acquiring the mobile phone division of Motorola for some \$12.5 Billion. Motorola was one of the pioneers of the mobile business so holds many patents that Microsoft and Apple will themselves be infringing. This coupled with Googles very deep pockets mean that the time when small manufacturers who have chosen to base their phones and tablets on Android no longer have to fear the mobster style shakedown. It is going to be interesting times indeed.

Why am I selling Tux Ties?

Martin Houston

[Advertisement]

I have owned a Tux tie since the late 1990s and the past couple of years people started to remark it was getting past its best!

First I just thought it would be 5 minutes with google and a fresh one would be on its way to me, but no, apart from going to Zazzle or Cafe Press and having one individually made at a premium cost.

So what I then did was find out what other clubs and societies did about their ties. I found Michelsons and they did me a new design with plenty of small Tuxen on it. Still saying 'Linux' but no longer having to shout so loud.

Free software such as Linux has grown up with the Internet and works because the benefits of sharing outweigh the costs of making copies. This is in stark contrast to most of the 'real world' economy where large amounts of money are needed to get things done. In short as a tie is in the 'real world' if you want one you have to hand over some money :) Although FLOSSUK members can have one at the special price of just £8 including VAT and postage instead of the normal price of £14.40.

What Linux lacks are things to sell. This would give the online world of Free Software and Linux some visibility in the commercial world. The simple mechanics of this is that money buys marketing and if that marketing makes more people look at Linux and other free software as mainstream, to be trusted, then that is good for all of us.

In the 'bad old days' before many people with reasonably fast broadband this used to be physical CDs and then DVDs. I myself was one of SuSE Linux's first UK resellers, just being a sideline from my Linux consultancy. I gave that in when I saw that Personal Computer World stores started stocking them, I felt my job was done at that point!

As Linux is so fast changing fetching ISOs and installing over the Internet makes better sense. There is still scope for books to be written. Haynes the car manual people do a very good one introducing Ubuntu Linux for example. However any printed book is going to be somewhat out of date compared with where the state of development is now. This is not always a problem, some parts of Linux are very stable, a 5 year old book will still tell you a lot of what you need to know, but if you are doing something at the 'bleeding edge'. The shorter publishing cycle of magazines makes more sense for Linux, notice the number of book/magazine hybrids that have come out recently.

Tux represents a very strong brand for Linux, with Linux here meaning the multitude of collections of free and open source software that collect around the Operating System kernel of the same name produced by Linus Torvalds and huge team.

Larry Ewing the creator of the original Tux logo has allowed the contented penguin to be used both on his own and as part of larger designs without the usual rigmarole of permission and approvals.

Anyway I emailed Larry and asked if he had any objection to me making some new Ties and he said 'go ahead' so now I have boxes and boxes of them!

I would also encourage other people to look at producing different Linux and Free Software 'collectables' for sale too. I for one would be a keen customer :) Having some physical items to show the sceptical will help getting other people to realise what a strong community we have.

<http://www.deluxetech.co.uk/tuxtie.html>

Asterisk: The Definitive Guide

Leif Madsen, Jim Van Meggelen and Russell Bryant

O'Reilly Media

ISBN: 978-0-596-51734-2

736pp.

£ 42.50

Published: May 2011

reviewed by Andy Thomas

Every complex system be it software or a hardware device deserves a 'Definitive Guide' exploring and explaining in depth each and every feature, command, configuration directive, syntax and so on and Asterisk is no exception. But in this, the third edition of Asterisk from O'Reilly, the authors attempt to combine this with a hands-on tutorial so a better title perhaps would have been 'Asterisk: The Book'. After reading the foreword and preface you soon realise that Asterisk is not just a 'Linux-powered PBX' – it is an incredibly flexible, extensible and capable technology that can do far more than any commercial office PBX but to a newcomer (even those with some traditional PBX and/or Linux experience) it appears at first sight to be formidably daunting to set up and maintain; this book sets out to get new users up to speed while at the same time serving as "the" reference book.

Asterisk has come a very long way since it first saw the light of day 12 years ago when a certain Mark Spencer, then running a Linux support business, found he could not afford a PBX for his office so set about creating his own using a Linux PC. At about the same time, an electronics engineer called Jim Dixon started the Zapata Telephony Project (Zaptel) with the aim of harnessing the power of the CPU in an ordinary PC to perform the functions of expensive, proprietary and closed hardware marketed by the big telecomms companies. Jim and Mark met and started to collaborate, with Jim's Linux Zaptel drivers finding their way into Asterisk to support his PCI-bus telephony line cards. It wasn't long before Mark's technical support operation decided to focus on Asterisk and try to break into the telecomms market and so Digium was born and started manufacturing their own analogue and digital telephony line cards for PCs.

Today, Asterisk has made serious inroads into the traditional proprietary PBX market dominated by the big players such as Nortel, Siemens, Panasonic, Lucent, Ericsson, GPT, et al and has been the main driving force behind integration of VoIP with traditional public telephone services and adding extra facilities such as video conferencing, follow-me, IM, emailing recorded conference calls as sound files to others and so on. Not content with this, it is now making inroads into the VoIP arena itself, challenging even Cisco.

Enough history, now onto the book: throughout the book the intended audience is a reasonably competent Linux user who is happy to set up a PC dedicated for Asterisk use and has a modicum of telecomms knowledge. Chapter one gives us an overview of the Asterisk project and its goals and finishes by introducing us to sources of further information, documentation and support. In chapter two we explore the architecture of Asterisk and it soon becomes clear why it is so successful – it is entirely modular in concept so it is easy for the Asterisk community (which is large and growing, by the way) and third parties to develop new modules that add functionality to the system. This modular approach stands traditional PBX thinking on its head; gone is the simple but rigid concept of internal

extensions and outside lines – Asterisk treats all devices (known as channels) equally so your desk phone, your mobile (cellphone for US readers), your fax machine, VoIP softphone and your external lines whether digital or POTS (Plain Old Telephone System) can all be integrated into a single system.

The remainder of chapter 2 throws us in at the deep end a bit as it is dominated by a long list of modules extending over several pages that might have been better placed later in the book when we really get stuck into Asterisk. This is one example of several where the authors' attempt to combine a tutorial with a definitive reference gives the book a slightly disjointed feel at times; hopping from a perfectly readable introduction to its architectural features and then plunging into an interminable list of modules (which may well be out of date by the time you read the book) before emerging back where the first part of the chapter left off to meet at last the core component of Asterisk – the Dialplan – means the book isn't exactly an easy read.

Assuming the reader hasn't been put off at this point, the tutorial side of the book begins to shine in chapter 3 where we first install Linux on a PC and then build Asterisk from source. To cater for the likely distributions of Linux that may be used on a Asterisk server, the detailed installation examples given are for CentOS and Ubuntu although there's no reason why Asterisk can't be used with other Linux distros (I used Debian 6 while reviewing the book) or with other UNIX-like operating systems. Experienced users might want to skip the first few pages and move straight on to building Asterisk but there is useful information there about dependencies and pre-requisites that should not be ignored.

Detailed instructions for building and installing Asterisk from source follow but if your PC has any Digium telephone line cards fitted, you should build and install the DAHDI-Linux or DAHDI-Linux-complete packages first before Asterisk as this will save a lot of time later when you discover Asterisk can't see your hardware! In fact, you are recommended to install DAHDI anyway even if you don't have any telecomm hardware interfaces in your PC as a few Asterisk features are dependent on DAHDI. This chapter is one of the best in the book, very clear and concise for what is a rather involved build sequence and ends with initial configuration of your newly-installed system. You can of course install your distribution's binary packages if you prefer but you won't get the latest version (currently 1.8.5.0 at the time of writing) nor will you get the same 'feel' for Asterisk as you would if you build it from source as the book recommends. This edition of the book covers all versions of Asterisk up to and including the latest version 1.8.

Chapter 4 is headed 'Initial Configuration' but didn't we just do that at the end of chapter 3? Well, yes and no – chapter 4 continues where chapter 3 ends and gives the impression it was written by a different author but nevertheless there are some more configuration files to set up before we can move on to user device configuration in chapter 5 where we set up our phones, VoIP softphones, etc as Asterisk channels. Finally, we create a basic dialplan to test the configuration so far and using the example shown, on dialling the number 200 on whatever device you have configured into your Asterisk installation, you should hear someone say "Hello world!".

The dialplan is where you perform most of the Asterisk user-level configuration and is covered in great detail in the next chapter while outside connectivity, where we connect our PC to your incoming analog or ISDN telephone lines, or to a SIP provider via your network connection, is the subject of chapter 7. By the time you have reached this part of the book, all being well you should have a functioning basic PBX capable of making calls to other devices connected to your Asterisk system and receiving incoming calls. Throughout the remainder of the book, all the other exciting features of Asterisk such as fax integration, voicemail, music on hold, integration with SQL databases, call queuing and many others too numerous to list here are covered, with a chapter devoted to each topic. New features are being added all the time so listing them here is rather pointless – and this is a review of the Asterisk the book, not Asterisk itself!

In the final Chapter 27, entitled 'Asterisk: A Future for Telephony', a play on the title of the 2nd edition of the book 'Asterisk: The Future of Telephony', the authors discuss how and why the telecomms industry has lost its way and is in real danger of going the way of the dinosaurs. In many ways the industry today is in a similar state to IT in the bad old days of 30-40 years ago, the days of mainframes and minis using incompatible hardware and running proprietary closed operating systems that were

equally incompatible with each other. It wasn't until the mid-1970's that cracks began to appear in the cosy vendor lock-in models – the emergence of CP/M as the first vendor-neutral universal microcomputer operating system (admittedly still closed and only usable on x80 processor systems) broke the ties between hardware and software vendors followed a few years later by the IBM PC which ultimately (and rather unwittingly from IBM's point of view) did the same for the computer hardware industry. Where are most of those 20th century vendors now? The same thing will inevitably happen in telecomms and as the book puts it, the industry must change, or cease to be.

Rounding off the book are three useful appendices describing telephony technology in general, VoIP protocols and on choosing a suitable hardware platform for your Asterisk server. Written with obvious enthusiasm by the three authors who are all heavily involved in Asterisk development, you cannot fail to grasp that Asterisk is big, is getting bigger and better and although not yet a household word, is a disruptive technology that puts the writing on the wall as far as the mainstream telecomms world is concerned.

Summing up, this book makes a brave and largely successful attempt at combining a tutorial with a very detailed reference but I have some minor reservations about the organisation and 'flow' of the book. I think most readers will be able to recognise this and find the book indispensable in getting an Asterisk system up and running but maybe O'Reilly should consider publishing this book additionally as a separate tutorial and as a reference book for those who prefer it this way as Asterisk is a large and complex subject. As the the authors themselves admit, the book only just scratches the surface.

Last but not least, the book has finally got me using Asterisk myself. While reviewing a book, where possible I like to try out or implement ideas from the book and I had always toyed with the idea of replacing the 20 year-old 6 extension PBX we have at home with an Asterisk PC-based system since my PBX does not support DTMF (tone dialling) phones and I could do with a few more extensions anyway. But I couldn't really justify the expense of PC analogue line cards – being specialist and not exactly mass-market items, they are not cheap. The book coming up for review was the catalyst I needed – within a day of it arriving, I managed to find a Digium PCI 4-port analogue line card on eBay for a reasonable price fitted with 3 FXS and one FXO modules (that's 3 extensions into one outside line in old-style PBX-speak), fitted it to a spare PC and was able to work my way through the book.

I soon discovered there is a thriving Asterisk community with a Wiki, mailing lists, user groups and online forums (I had a problem getting the DAHDI channel driver to recognise pulse-dialled digits from my collection of old GPO telephones with rotary dials and received responses to a forum post within hours). If you want to try something different with FLOSS, and breathe new life into your dull old landline, try Asterisk!

The Book of PF

Peter N M Hansteen

No Starch Press

ISBN: 978-1-59327-274-6

216pp.

£ 23.99

Published: November 2010

reviewed by Andy Thomas

Based on the original BSD IPFilter, which itself lives on in Solaris 10 and later, PF first saw the light of day in 2001 largely as a result of the unexpected discovery that the original BSD license covering IPFilter did not actually permit the modifications and extensions the OpenBSD community had been making to their version of the IPFilter source. IPFilter was hurriedly withdrawn from the then current OpenBSD release but fortunately for everyone, at about this time Daniel Hartmeier had been doing some experiments of his own with a modified network stack that allowed him to insert his own code through which packets could be diverted. The removal of IPFilter from OpenBSD served as a powerful impetus to accelerate development of Daniel's ideas and less than a month after OpenBSD lost IPFilter in May 2001, PF was committed to the OpenBSD project and following intensive development during the remainder of that year, became part of the OpenBSD 3.0 base distribution in December.

Packet Filter (PF) has become arguably the most comprehensive and robust network control software out there which also happens to be free. It has since been ported to the other open source BSDs (FreeBSD, NetBSD and DragonFly BSD) although the most recent version of PF will always be found in OpenBSD. Compared with Linux iptables, PF rules are much more readable and make logical sense, at least for English-speaking users, which to some extent makes them self-documenting.

Reflecting PF's capabilities, the Book of PF does full justice to such a powerful tool and throughout the book, examples of using PF on OpenBSD, FreeBSD and NetBSD are presented to cover the slight differences between these operating systems; DragonFly BSD users will nonetheless find it easy to work with this book. With its beginnings in a lecture to a BSD user group in Norway, the original document evolved into a tutorial and then became the first edition of this book; this second edition, released earlier this year, has been extensively updated and revised.

After a quick orientation on the BSD way of doing things (for the benefit of those coming from other operating systems), the reader is taken straight into chapter 2 where we enable PF on a BSD system, write a simple pf.conf file and then check and test it. At this early stage, the use of macros in the pf.conf configuration file is introduced – this makes the configuration much more readable and easier to understand. Also, the concept of checking and thoroughly testing every rule is dinned into the reader; when you think about it, this makes an awful lot of sense when you remember the likely applications of PF – remote firewalls and routers in inaccessible places, servers in remote data centres and so on. A mistake in a PF rule could look you out of a remote connection leaving you with a broken or incomplete firewall, a downed server or worse, leading to red faces and long trips away from home!

Reaching chapter 3, we are ready to emerge into the real world and here we create a network gateway (or router, if you like). Tips abound on making the gateway easier to troubleshoot, using tables in the pf.conf configuration file to make it easier to maintain and handling the special requirements of “that sad old FTP thing” as the author so drolly puts it (that phrase really had me in stitches for a while).

Wireless LANs get extensive treatment in chapter 4 and we are shown how to create both infrastructure wireless access points (WAP) and set up client systems using both OpenBSD and FreeBSD. Most readers like myself will have loads of spare computers lying around so the idea of building your own access point using BSD and PF will appeal to many, since control over most consumer-type WAPs is very limited. Also well-covered in this chapter is the use of authpf on OpenBSD to create an authenticating wireless gateway in addition to (or even instead of) WEP/WPA security.

Chapter 5 takes us into “big boys” territory – DMZs, synproxy, load-balancing and relayd – but fear not, the author explains everything clearly with plenty of examples. Packet tagging using the PF tag

keyword is explored before we move onto raw Ethernet country and build an Ethernet bridge which demonstrates PF has plenty of applications outside the TCP/IP world. Proactive defence mechanisms including adaptive firewalling to thwart brute-force attacks and the OpenBSD spamd (not to be confused with the SpamAssassin daemon of the same name) as a comprehensive and very agile defence against unwanted email are covered in chapter 6 while the next chapter introduces us to ALTQ (ALTernative Queueing), hsfcr and friends in the network traffic arena. CARP (Common Address Redundancy Protocol) and pfsync, used to keep pairs of redundant firewalls in sync, are encountered towards the end of this chapter; this had particular interest for me as I stood on the sidelines of a failed attempt by others to implement CARP on a pair of firewalls last year and armed with the knowledge I have gleaned from this book, I may well have a go at this myself.

Monitoring and collecting packet statistics follows in the penultimate chapter 8 and we soon meet pflow, a pseudo interface that can be used to collect NetFlow data and pass it on to a network flow analysis system. The information presented here integrates very well with Michael Lucas's book "Network Flow Analysis" from the same No Starch stable so those with a serious interest in networks will do well to purchase this book also. The final chapter contains tips on debugging PF rulesets, what to tweak (and what's best left well alone) and various other snippets of general information that don't quite fit into earlier chapters. Two appendices round off the book – the first containing references to sources of more information and the second discussing hardware that is compatible with OpenBSD – and finally an index.

Written in a conversational style that immediately warms the reader to the author as if we are discussing PF over a coffee, it contains some memorable phrases such as when an Ethernet bridge is defined as "an intelligent bulge on the network cable, which can filter and redirect". It is not without humour either, as when the author gives the FreeBSD PF home page maintainer a gentle dig by saying "it will hopefully spring into life again once [its maintainer] notices he is referenced in a printed book". As one reviewer of the first edition declared, "This book is a super-easy read. I loved it! This book easily makes my Top 5 Book list" and I have to agree.

I personally found this book useful having being thrown into PF at the very deep end last year after I took over FreeBSD support for a web hosting company and a medium size business infrastructure from a company that had closed down; it wasn't long before I discovered they all had OpenBSD PF firewalls as well which were now my responsibility. Now, I had tinkered with ipf before on a Solaris NAT gateway at home but the industrial strength pf.conf's on these Open BSD firewalls were something else – lots of VLANs, multiple VPNs, virtual interfaces, port forwarding tricks, etc, you name it, these firewalls had it! Whoever had written these PF rulesets really knew his stuff but of course, it was hardly documented as usual. Panic began to set in when some of these firewalls started to suffer hardware failures, rule changes were requested, etc and finding Peter Hansteen's book on Amazon was a real godsend.

In summary, this book is not a cookbook of PF rules nor is it a 'howto' – indeed the author cautions against using it in this way as the PF rules given in the book taken in isolation may not have the desired effect if cut and pasted into some other ruleset. His intention is for the book to help you understand what each rule does, that you have a clearly defined objective for your ruleset, that you check and test your rules and you take the trouble to understand your network. This book will give a competent system or network administrator the confidence to set up, modify or fix a PF installation without the fear he/she may be phoned up a few hours later in the middle of the night to be told some remote branch office has disappeared off someone's corporate network. And that does happen!

Javascript: The Definitive Guide

David Flanagan

O'Reilly Media

ISBN: 978-0-596-08552-4

1098pp.

£ 38.50

Published: May 2011

reviewed by John Collins

JavaScript is the scripting language used to provide some interactive facilities on web pages. Programs written in JavaScript are executed by the client browser client rather than the server. This can provide some considerable interaction with the user before any transactions are executed between the client and the web server. Typical uses are for checking forms, providing “shopping baskets” and similar. New windows can be created and contents filled and updated entirely from a starting page with little or no interaction with the web server.

Some email clients can support it also but most users turn this off as they don't want unexpected browser windows or similar appearing when they are reading emails.

Despite its name, JavaScript has little connection with Java. The syntax is similar in many places (and derives from C), but unlike with Java, it is not “strongly typed”. The type of each value is determined entirely at runtime. This may of course give bugs which the strict type checking of Java would eliminate at compile time.

This book claims to be “The Definitive Guide” and it is hard to argue with that description. I have owned 2 of the 3 previous editions and could not have managed to do anything serious in programming web pages without them.

The book introduces JavaScript and explains the syntax and facilities, gives extensive examples of programming techniques and their effects and then gives a two part reference of JavaScript functions and facilities and then the attributes of HTML elements which can be manipulated from within JavaScript.

The new edition is even more massive than before, containing a good deal more detail in some places and also describing some of the newer features of ECMAScript 5 and HTML 5 (ECMAScript is a previous name for JavaScript). As ever, the user cannot rely on these features being generally available on some arbitrary user's browser but this book is careful to make clear what is going on.

This book is as helpful, comprehensive and well-written as it is possible for any book to be. You cannot possibly do anything serious in terms of programming web pages without it. Even if you've got a previous edition, I think you'll probably want to get this one too as the explanation of how to do some things, for example mouse events and interaction with CSS is much better than before in my view.

Agile Web Development with Rails

Sam Ruby, Dave Thomas and David Heinemeier Hansson

Pragmatic Bookshelf

ISBN: 978-1-934356-54-8

480pp.

£ 33.99

Published: April 2011

reviewed by Paul Waring

This book, a well-known introductory guide to Ruby on Rails, is now in its fourth edition, so most problems have been shaken out by this point. It is also reasonably up to date, covering Rails 3 which was released a year ago. Assuming it stays around for a similar amount of time to the previous version, readers should be able to get at least three years worth of value from this book.

The first chapter covers the installation of Rails on the usual Windows, OS X and Linux platforms. What the guide unfortunately fails to mention is that if you install Rails from a package repository (as I did by default) you will probably find yourself with version 2.x, which will not be compatible with some of the examples and tutorials in the book! Some suggestions of editors are also made, which is useful as not all IDEs will support code completion for Rails.

After installation we have the obligatory 'Hello World' chapter, which covers the basic commands such as creating a new project and starting the WEBrick development server. A few basic changes to the example, including the introduction of variables and linking to other pages are covered, and by the end of the chapter the reader has a basic application up and running. A relatively short chapter then follows on the Model View Controller (MVC) pattern, which readers may want to supplement through online resources as the text does not go into much detail.

The next chapter is a basic introduction to Ruby, and a sensible addition given that many people who are new to Rails are also new to Ruby (unlike Catalyst, where a reasonable knowledge of Perl can be assumed). All the basic concepts are covered, and for most readers it will be a case of getting used to the Ruby syntax as opposed to approaching constructs such as loops for the first time. Common Ruby idioms are also discussed, which might otherwise confuse programmers new to the language, and links are provided to more comprehensive online lists.

The rest of the book is dedicated to developing the archetypal shopping cart application, introducing a new concept in each chapter. Whether this is a good thing or not is a matter of personal choice. Some readers will find the step by step development of a complete application a useful way to pick up a new language/framework, whereas others, myself included, prefer concepts to be explained in a more generic manner, so we can quickly apply them to different projects.

The only thing which slightly annoyed me about this book, and is a problem with the whole Pragmatic Programmers series, is the typesetting. A better choice of font would make the reading experience much more enjoyable. However, even with that minor issue, this is a well-rounded introduction to Ruby on Rails, and worth a read if you want to approach the framework as a programmer experienced in other languages but new to Ruby.

Learning Android

Marko Gargenta

O'Reilly Media

ISBN: 978-1-4493-9050-1

268pp.

£ 26.99

Published: March 2011

reviewed by Paul Waring

Although the iPhone is probably better known outside technical circles, Android handsets have been quietly gaining ground in the past few years, spurred on by a more open approach to application development. Being a big fan of Learning Perl and Learning Python, and looking to develop Android applications, I had high expectations for this book.

The first two chapters present an overview of Android, which is particularly useful when the name is used to describe a variety of parts (sometimes the whole platform, sometimes just the operating system). The whole stack is explained in a useful diagram which demonstrates the various layers from the kernel up to the applications. The security of Android is glossed over with, to paraphrase, a 'Linux is secure, Android is built on Linux, therefore Android is secure' argument, which is a bit disappointing but perhaps to be expected in early chapters.

The next chapter is the traditional quick start and Hello World introduction. Installation instructions are fairly limited, and the author assumes that the reader is using the Eclipse IDE, so if you prefer a different editor you will be on your own. However, Eclipse is available for every major platform, and it makes sense to demonstrate one popular IDE in detail than vaguely discuss how applications might be built in a variety of environments.

Chapter four covers the main building blocks of applications, followed by a chapter giving a brief overview of the application which will be used as the example for the rest of the book. In chapter six we finally get down to some coding with the development of an application for accessing Twitter which crams in user interfaces, threading and network access. The next chapter is similarly packed, covering user preferences and the filesystem, although with only a tiny section on filesystem security. By chapter eight I must admit I was starting to lose interest, and whilst I made it to the end the process felt like a lot of effort for little reward.

Overall, this book leaves me feeling a bit disappointed with the lack of detail. Some chapters are very short, often only a few pages, whereas others try to cram too much in. Security and testing are also two topics which are conspicuous in their absence. I would prefer fewer and more consistently sized chapters, with more detailed explanations, to a whistle-stop tour which leaves me none the wiser at the end. If you are looking for a high level overview of Android then this may be the book for you, but I would not recommend it as the place to start if you want to develop applications for the platform.

The Book of CSS3

Peter Gasston

No Starch Press

ISBN: 978-1-59327-286-9

304pp.

£ 27.49

Published: May 2011

reviewed by Lindsay Marshall

I was going to say that books about HTML5 and/or CSS3 are two a penny at the moment, but given the price of books that is an inadequate analogy. Let's just say that there are a lot of them. And the standards are still not entirely settled so lots of opportunity for second (and further) editions when they do get made properly official and browsers implement all the features... Now, I bet you are thinking that is a typical introduction to one of my cynical book reviews (which assumes that you a) read reviews, and b) look at the name of the person who wrote them, but never mind), but you would be wrong.

I love this book. I think it is fantastic. Best book on CSS of any kind I have seen for ages and ages. It is promoted to a place on my desk as of now. Full of sensible, clear examples. Lots of useful information about browser support and differences in implementation. All the alternate syntax information you need to get your CSS files right. What else do you need? (Apart from fully compliant browsers and everyone using them that is).

Nicely produced, well written. Yes, it will get out of date quite quickly, but if you want a leg up for CSS3 this is definitely it. It's pricey but not insanelly so. Definitely one for the team library if not for your desk.

iOS4 Programming Cookbook

Vandad Nahavandipoor

O'Reilly Media

ISBN: 978-1-4493-8822-5

640pp.

£ 38.50

Published: February 2011

reviewed by Mike Smith

A nice big book this one – around 600 pages and an extensive index. The traditional cookbook style with a myriad of “recipes” categorised into 15 chapters. The topics range from using the Objective C language, various cocoa frameworks, how to implement multitasking, networking, interacting with iOS device hardware and more.

Whilst the book is not aimed at the beginner, and it states as such in the introduction, there are a number of recipes that are useful to those new to the iOS environment. These help clarify how and why you undertake certain activities – such as how to add a new class to your project, and for those new to object-oriented programming how to emulate C function calls with objects (using [self]). So it is definitely suitable for the new iOS coder who's early in the learning curve.

Each recipe is composed of a “problem”, the solution and a discussion. There are plenty of code snippets and where relevant screen shots too.

One of the recipes has a fairly mundane title of “managing master-detail views”. A more user-friendly way of saying this might have been “how to tell if I'm running on an iPad”, as the technique tests to see if we can use a `UISplitViewController`. To be fair it does go on to show how to use the

class, but I thought that the initial test would be a useful recipe in its own right. There is a more general point though that perhaps some recipes would have benefited from having more descriptive titles (and less linked to the iOS jargon). It's a minor point.

Other recipes show how easy it is to, for instance, show maps in an iOS device – using both interface builder and programatically, and then use Core Location to find your location and drop pins on the map. I didn't realise MapKit could perform latitude/longitude coordinate to (postal) address translations too. Very useful.

I liked the way some recipes built upon previous ones; building up more sophisticated mini-apps. There's a companion website too. So far so good.

For me the format of these cookbooks works. It permits you to dip in and out quickly as a reference text. If pressed to find fault then I have two observations. Firstly, some of the recipes are very basic – just examples of how you use specific classes and little more than you'd find in the SDK documentation. The value in this book is when you apply techniques to practical problems. Secondly, of course there are only so many recipes you can fit in a book like this – so I'm afraid there's a good chance you won't find what you're looking for. I am currently developing a little personal application (for controlling my central heating from the iPhone). There are several (probably quite common) things I'd like to do to enhance what I've done so far and I would have liked to see some of them in this book. I'll give you some examples. For instance checking that a view is actually displayed before attempting to refresh the data in it when the app returns from the background. Also techniques to avoiding invoking a second action when one is already running (instance variable I suppose, but perhaps there's a built-in way within iOS). There is similar concepts (and very useful too), but not quite these particular ones. A third example would be handling the keyboard – changing the button labels and actions, moving the view so fields are not covered by the keyboard thereby ensuring they are editable etc). So I didn't see material covering these topics, but in other areas the examples have prompted me to think of alternative ways to address a particular requirement. For instance I wanted to make an image respond to a touch-up event – I haven't identified how I should do this yet, but I do now know how to handle tap gestures in the `UIView`.

So in summary I'd say that this is a good book and worth you having a look at. There's scope for future enhancement, or even a volume 2 if O'Reilly could be persuaded, as I am sure there are many more techniques that could be added to create a whole series. Check it out.

The author lives and works in Blighty too, so a resonance with the UKUUG.

Contributors

John Collins is a former UKUUG Council member and runs Xi Software Ltd which specialises in System Management tools for UNIX and Linux systems. Xi Software is based in Welwyn Garden City and has been in existence for about 25 years.

Martin Houston has been using Unix since 1980 starting with a PDP11/70 and Linux since 1994. He was the founder of the original UKUUG Linux SIG and is currently a freelance Linux consultant. He lives in Essex with wife Teresa and 2 cats, 2 dogs and too many computers.

Lindsay Marshall developed the Newcastle Connection distributed UNIX software and created the first Internet cemetery. He is a Senior Lecturer in the School of Computing Science at the Newcastle University. He also runs the RISKS digest website and the Bifurcated Rivets weblog.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Mike Smith works in the Chief Technology Office of a major European listed outsourcing company, setting technical strategy and working with hardware and software vendors to bring innovative solutions to its clients. He has 20 years experience in the industry, including mid-range technical support roles and has experience with AIX, Dynix/ptx, HP-UX, Irix, Reliant UNIX, Solaris and of course Linux.

Andy Thomas is a UNIX/Linux systems administrator working for Dijit New Media, Imperial College London and a number of small companies. Having started with Linux when it first appeared in the early 1990's, he now enjoys working with a variety of UNIX and Linux distributions and has a particular interest in high availability systems and parallel compute clusters.

Paul Waring is chairman of UKUUG and a director of a wholesale insurance broker. Outside of work he can usually be found filing documentation bugs against various open source and free software projects.

Roger Whittaker works for SUSE supporting SUSE Linux Enterprise Server for major customers in the UK. He is also the UKUUG Newsletter Editor, and co-author of three successive versions of a SUSE book published by Wiley.

Contacts

Paul Waring
UKUUG Chairman
Manchester

Phil Hands
Council member
London

Martin Houston
Council member
Chelmsford

Kimball Johnson
Council member
Lancaster

Holger Kraus
Council member
Leicester

Niall Mansfield
Council member
Cambridge

John Pinner
Council member
Sutton Coldfield

Howard Thomson
Treasurer; Council member
Ashford, Middlesex

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273475
Fax: 01763 273255
office@ukuug.org

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Manchester