INSIDE: Deploying Open Source Databases - MySQL

o3:

The Open Source Enterprise Data Networking Magazine

Issue 5
August 2006

http://www.o3magazine.com

TurboGears Author Kevin Dangoor provides a Quick Look at TurboGears the Python agile web platform

WiFi Security
Threats from the inside

Introduction to Python

ATA over Ethernet

Deploying T1s with
Linux and Sangoma A10x cards

Layer 2 Security Testing
with Yersinia

TurboGears: Deployment and Scaling

“No matter where in Europe your customers are ....

IKON COMMUNICATION SERVICES

.... Internet performance won't be a problem."

Your business gateway to Europe

Plug your business into the European market with Ikon's advanced network services.

Fast: Ikon Communications has a state of the art network with full transit from Europe's leading providers. Providing some of the fastest possible speeds to practically anywhere in Europe.

Flexible: Whether you need colocation, dedicated servers, web hosting or other services. Ikon's friendly and knowledgable team will help you provision what you need and when you need it.

Local: Ikon operates from the London Docklands area, in state of the art datacenter facilities. Placing local services close to your European customers not only provides faster access but provides an edge over your competitors.

+44 870 141 7273

"Simply a better network..."

sales@ikcs.net

WEBSITE
http://www.turbogears.org

LICENSE
MIT / LGPL / Other Open Source

AUDIENCE
Developers / Web Developers

OVERVIEW
TurboGears is a rapid web development megaframework with the goal of creating great web applications faster.

12 Introduction to Python

Python is a dynamic object-orientated programming language. TurboGears is based around Python, so a working knowledge of Python is key to getting the most out of TurboGears.

17 A Quick Look at TurboGears

TurboGears Author Kevin Dangoor provides a quick introduction to TurboGears, how it works and how to get started.

25 Deployment and scaling

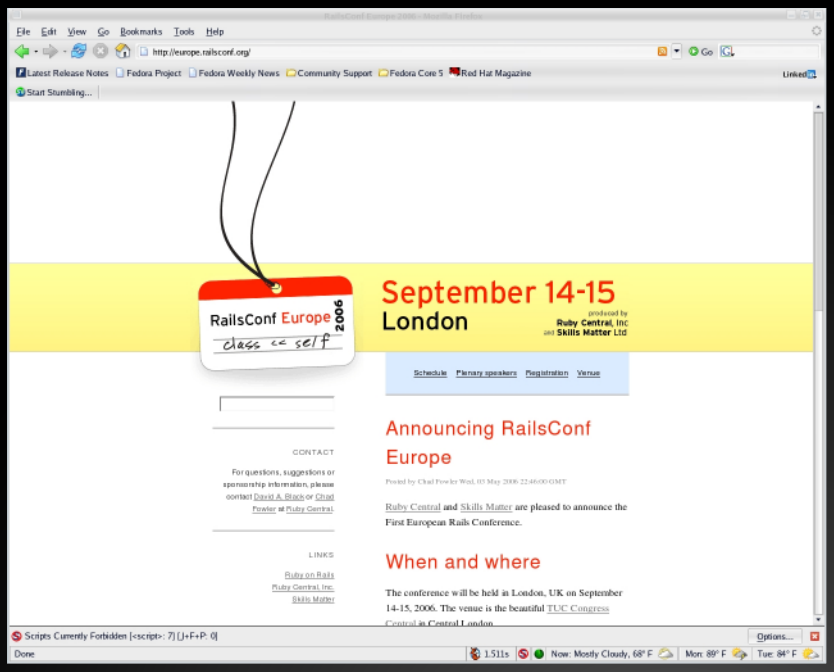
Creating great web applications in a quick and easy fashion is great for web developers, but will the resulting solution deploy easily and scale as well as a traditional web application ?

Issue 5
August 2006

Security

Networking

Focus on Databases



WEBSITE
http://europe.railsconf.org

VENUE
London, UK

WHEN
September 14 - 15, 2006

SPEAKERS
David Heinemeier Hansson
Dave Thomas
James Duncan Davidson
Jim Weirich
Kathy Sierra

Upcoming Events

YAPC::Europe 2006
http://www.birmingham2006.com

Birmingham, UK
Aug 30 - Sept 1, 2006

The Yet Another Perl Conferences (YAPCs) are grassroots symposia on the Perl programming language.

Akademy 2006
http://dot.kde.org/1142439906/

Dublin, Ireland
Sep 23 - 30, 2006

The annual KDE World Summit, Akademy, will be held at Trinity College, in Dublin.

LinuxWorld Russia 2006
http://www.linuxworldexpo.ru

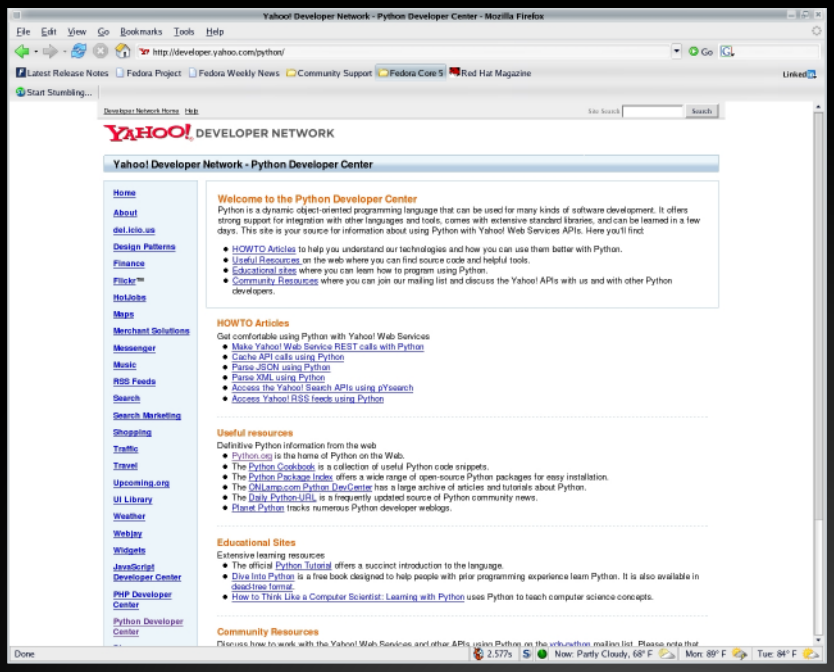
Moscow, Russia
Sept 4 - 15, 2006

Linuxworld Russia is the most famous event on the Russian market and is the meeting point for everyone who is interested in open source solutions in Russia.

Ohio LinuxFest 2006
http://www.ohiolinux.org

Columbus, Ohio, USA
Sept 30, 2006

Ohio LinuxFest is a free annual conference and event for the Linux and Open Source community. Hosting authoritiative speakers and a large expo, the event welcomes professionals and enthusiasts.



WEBSITE

http://developer.yahoo.com/python

DETAILS

Yahoo! recently launched a new Python Developer Center as part of the Yahoo! Developer Network.

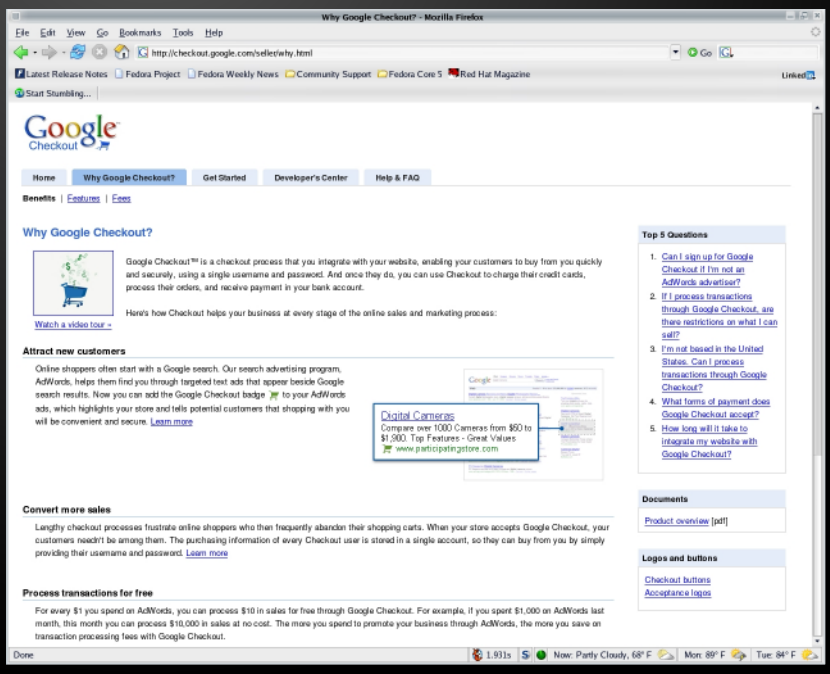
The new resources is primarily to help Python developers to interact with Yahoo! Web services. The Yahoo! team have done a good job rounding up excellent general purpose Python resources. Whether you're new to Python or not, it is a good reference site for Python developers.

WEBSITE

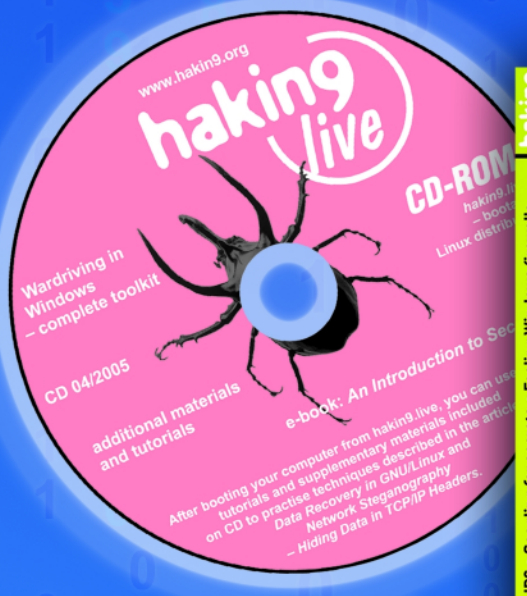
http://checkout.google.com/

DETAILS

Google Checkout (tm) is a checkout solution that can easily be integrated into any website. Google have a pretty slick solution that is also integrated into their AdWords service. The integration is in the form of a small cart symbol next to AdWords. With fees as low as 2% + $0.20 per transaction, and discounts for AdWords customers. Google Checkout is a very competitive offering.



We have knowledge. Want some?



www.hakin9.org/en

If you want to buy a magazine, please visit us at www.shop.software.com.

available at the beginning of July

Free Software MAGAZINE

The free magazine for the free software world

- Articles are released under a free license
- Available online as HTML or PDF
- Packed with amazing content
- Both technical and non-technical articles

GO AND SEE FOR YOURSELF!

WWW.FREESOFTWAREMAGAZINE.COM

o3 magazine has returned

AFTER A SHORT BREAK, NEW HARDWARE, NEW SERVERS AND A NEW RELEASE OF SCRIBUS WE HAVE CONTINUED THE MAJOR OVERHAUL OF O3 AND WE HOPE YOU LIKE THE RESULTS...

By John Buswell

Welcome to Issue 5 of o3 magazine. As you have probably noticed, this issue is several months late. Earlier this year, I was forced to step back from o3 due to personal reasons. My wife had some serious complications towards the end of her pregnancy, which resulted in delegating most of o3 to a new and inexperienced team. The good news is that my wife and 4 month old son are healthy and doing great. I've also learned the hard way that while delegating is good, delegating to the wrong people isn't the best way to keep a magazine running.

I am pleased to announce though, that we have a very aggressive schedule in place for catching up. We plan to have caught up to our monthly release schedule by the end of the year. Thats right, 9 issues between now and the end of the year. We have most of the content ready to go, so you can expect an active release schedule for the magazine over the last few months of 2006.

o3 is now produced using Scribus 1.3.3.3. I would like to thank the Scribus team for a solid release, which has helped us put o3 together a lot faster this month. We're now using Scribus under Fedora Cora 5, with a 5 LCD panel configuration utilizing 3 video cards. Our workstations are AMD Athlon64 X2 based, and since the move away from Mandriva, all of our performance problems have gone.

I would like to thank Kevin Dangoor for his help in putting together this TurboGears focused edition of o3 magazine. TurboGears is an excellent web application framework based on Python, definitely well worth a look. Over the coming months we will be comparing Rails, TurboGears and several other frameworks.

This issue marks several key content changes for o3. We have dropped the regular columns for a more feature orientated system. The new system will group several articles together under the main feature for the issue, this month its TurboGears. The new “Focus On” column is a multi-issue feature, for the next few issues, we will be focused on Open Source Databases. Security and Networking columns will cover both wired and wireless solutions.

Issue 6 which will be hot on the heels of this issue, looks at building IT infrastructure from scratch on a small business budget. The entire issue is a step by step guide to building a complex switched network providing both office workstations and server access.

Ohio LinuxFest 2006 is rapidly approaching, I will be at this event, at the Spliced Networks booth. So if you'll be near the event on September 30th, please feel free to stop by the booth.

o3 magazine
http://www.o3magazine.com

John Buswell
Publisher and Editor In Chief

Greg Jordan
Managing Editor

Cover Graphic
The TurboGears Golden Gear is the property of Blazing Things LLC. TurboGears is a trademark of Kevin Dangoor.

Publisher Information
o3 magazine is published and distributed by Spliced Networks LLC. o3 magazine is a trademark of Spliced Networks LLC. All other trademarks belong to their respective owners.

Reclaim lost time

Levanta Intrepid M™

The world's first Linux management appliance

The Intrepid M™ appliance performs the most important Linux management tasks in a fraction of the time you now spend. You'll save both time and money with a single point of control.

The Intrepid M does these tasks and more:

- **Quickly provision servers** — in minutes—even from bare metal
- **Easily deploy software and patches** – eliminating lengthy installations
- **Efficiently migrate software** - from one piece of hardware to another
- **Instantly rollback** - undo changes in minutes, even to the distant past

"Levanta reduced an eight-hour process down to five minutes."

-Arty Ecock, Manager of VM Enterprise Systems, City University of New York (CUNY), Computing and Information Systems

Download the Analyst Report

Levanta Marries Provisioning with Data Virtualization

www.levanta.com/downloads

LEVANTA®
www.levanta.com
1.877.LEVANTA

WINNER
LINUXWORLD
Most Innovative Hardware Solution

introduction to python

PYTHON IS THE PROGRAMMING LANGUAGE THAT TURBOGEARS IS BASED AROUND
IT IS RELATIVELY EASY TO LEARN AND THIS ARTICLE WILL GET YOU OFF TO THE RIGHT START

By John Buswell

Python is a dynamic object-orientated programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code. Python is distributed under an OSI-approved open source license, making it free to use even for commercial products.

Platforms

Python runs on Windows, Linux/Unix, MacOS X, OS/2, Amiga, Palm Handhelds and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines.

Python vs. Ruby

You have probably at least heard of Ruby, which is another language that is very similar to Python. There differences between Ruby and Python are small compared to the features of both languages. In fact if you learn one, moving over to the other is pretty simple. So if you already know Ruby, why look at Python? If you know neither, why learn Python, Ruby has that cool Ruby on Rails framework right? Well it doesn't hurt to know another programming language, so why not simply learn both? Which you choose is entirely up to you, and thats the nice thing about open source, you have a choice. If you search Google for “difference between python ruby”, you will get a lot of hits. One argument you'll see that might tip you in one direction or another is the claim that the Object Orientated features in Python are just “bolted on”, while Ruby is fully Object Orientated. This particular argument focuses on a really old version of Python, these days Python and Ruby are pretty much on par with each other. Python has enjoyed a little more popularity over the years, so Python sometimes has a better selection of libraries. You are less likely to run into problems that you find on RubyForge, where projects sound good but there is no code yet, with Python. Python 2.5 will have some performance optimizations, following a Need For Speed event in Iceland earlier this year that focused on improving the performance of Python.

Probably the biggest argument for Ruby is Rails. However, if we all made decisions on popularity at a specific time, you may have never bothered to look at Linux because Windows was more popular? In this issue, we look at TurboGears, one offering in the Python world that competes with Rails. One thing in favor of TurboGears is that it uses many seasoned open source projects as components, while Rails reinvents the wheel in many cases. Keep in mind that Rails is developed by essentially, web developers. The Rails project showed is inexperience earlier this year when a security problem is removed and reintroduced, then sorta fixed and then sorta half-announced. I don't really see the point in not disclosing the security problem, when you tell the public it disappeared and was reintroduced. Perhaps they haven't heard of the unix diff command? This type of project immaturity should make developers examine the solution to see if it meets their needs before jumping on the popularity band wagon.

Building Python

Due to some production delays, we've had time to update this article to reflect Python 2.5. Python 2.5 was released in September, but this is the August issue of o3. Python can be obtained from http://www.python.org.

```
$ tar jxvf Python-2.5.tar.bz2
$ cd Python-2.5
Python-2.5]$ ./configure \
–prefix=/home/johnb/projects/python/2.5/

Python-2.5]$ make
Python-2.5]$ make install

Python-2.5]$ cd ~/projects/python
$ PATH=/home/johnb/projects/python/2.5/bin/:$PATH
$ export PATH
$ which python
~/projects/python/2.5/bin/python
$ python -V
  Python 2.5
$ ls 2.5/
   bin include lib man
```

Hello World

While you can write python programs in files, embed them in other languages such as C and so on, whether you're starting out or a seasoned python developer, it always helps to execute some code and see the result. You can use the python interactive interpreter by simply typing python.

```
$ python

Python 2.5 (r25:51908, Sep 29 2006, 01:11:26)
[GCC 4.1.1 20060525 (Red Hat 4.1.1-1)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.

>>> print "hello world"
hello world

>>> x = 1
>>> y = 665
>>> print x+y
666
>>>
```

Feed the snake

Okay, Hello World isn't that impressive. Lets take a quick look at what you can do with python in a few lines of code, below we check to see if eth0 and eth1 are up.

```
>>> import fcntl, struct, sys
>>> from socket import *
>>> SIOCGIFFLAGS = 0x8913
>>> tmp256 = '\0'*256
>>> ifname="eth0"
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> rc = fcntl.ioctl(s.fileno(), SIOCGIFFLAGS, ifname +
tmp256)
>>> flags, = struct.unpack('H',rc[16:18])
>>> up = flags & 1
>>> print ('DOWN','UP')[up]
UP
```

The eth0 interface is up, what about eth1 :

```
>>> ifname="eth1"
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> rc = fcntl.ioctl(s.fileno(), SIOCGIFFLAGS, ifname +
tmp256)
>>> flags, = struct.unpack('H',rc[16:18])
>>> up = flags & 1
>>> print ('DOWN','UP')[up]
DOWN
```

So, was python correct, lets check ***ip link*** :

```
2: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc
noop qlen 1000
link/ether 00:xx:xx:xx:xx:xx brd ff:ff:ff:ff:ff:ff

3: eth0: <BROADCAST,MULTICAST,UP,10000> mtu
1500 qdisc pfifo_fast qlen 1000
link/ether 00:xx:xx:xx:xx:xx brd ff:ff:ff:ff:ff:ff
```

Sure was.

Going Further

So Python is starting to look interesting to you, how do you learn more? The first thing you might want is an editor, although your favorite text editor should work just fine, you can find a comprehensive list of Python related editors at http://wiki.python.org/moin/PythonEditors.

Whats better than editors, a good IDE. IDEs are integrated development environments, Eclipse is a good one thats free, and there is Python support. A full list of IDEs can be found at http://wiki.python.org/moin/IntegratedDevelopmentEnvironments.

Books are a great place to go, books will give you everything that you need to get started as fast as you can read it. If you're familiar with industry standard object oriented concepts such as classes and methods, then learning Python can be done in a few hours to a few days. Python actually has a number of free books, I personally like Dive Into Python, a full list of books both free and commercial can be found at http://wiki.python.org/moin/IntroductoryBooks. O'Reilly's Learning Python and Python Cookbook are handy additions to any Python programmers library.

Cool tools

One Python-centric tool is called Scapy, which can be found at http://www.secdev.org/projects/scapy/. It is a packet crafting tool, written in Python. It enables you to tweak a packet just the way you want it and then send it how you want to. Its a very very cool tool.

The Python Cookbook over at O'Reilly's ActiveState Programmer Network, which can be found at http://aspn.activestate.com/ASPN/Cookbook/Python/ has a wide range of code recipes from beginner to advanced. Its definitely a good place to learn how to do things, and a great reference as well.

IF statements

Before we wrap up we will take a quick look at some language basics with Python, starting if statements. In python you use if <test> : elif <test2>: else:

```
Python 2.5 (r25:51908, Sep 29 2006, 01:11:26)
[GCC 4.1.1 20060525 (Red Hat 4.1.1-1)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> x = 1
>>> y = 2
>>> z = 3
>>> if x:
...  y += 1
...  z -= 2
...  print x+y+z
... else:
...  print 'not x'
...
5
```

Here we set x.y and z variables, if x has a value, we added to y, and took away from z, and printed the result, otherwise we print "not x". The end result was 5 (1 + 3 + 1).

While loops

A while loop basically runs a block of code continously until a certain condition is met. Below we set p and q, add to one, subtract from the other until p is no longer greater than q.

```
>>> p = 20
>>> q = 0
>>> while p > q:
...  print p
...  print q
...  p -= 1
...  q += 1
...
20
0
19
1
18
2
17
3
16
4
15
5
14
6
13
7
12
8
11
9
>>> print p
10
>>> print q
10
>>>
```

For loops

Python has for loops too. The notation is very simple, its for <target> in <object>: <code_block>. Here is an example which takes a group of subnets and creates an ip route string for them all using Python, might as well demonstrate something useful :

```
>>> subnets =
["10.1.2.0","10.1.3.0","10.10.20.0","10.10.30.0"]
>>> gw = "192.168.1.20"
>>> notation = "/24"
>>> dev = "eth0"
>>> for x in subnets:
...  print 'ip route add ' + x + notation + ' via ' + gw + '
dev ' + dev
...
ip route add 10.1.2.0/24 via 192.168.1.20 dev eth0
ip route add 10.1.3.0/24 via 192.168.1.20 dev eth0
ip route add 10.10.20.0/24 via 192.168.1.20 dev eth0
ip route add 10.10.30.0/24 via 192.168.1.20 dev eth0
>>>
```

Conclusion

Python is a very useful language, as you've seen in just a few lines of code, we've done some pretty neat little things with it. This is just a very basic look at Python, it is capable of much more. Whether you choose to use Python is up to you, but you now have a least a very basic understanding to be able to evaluate platforms such as TurboGears a little better. In the end, the choice is up to you, I use both Python and Ruby on a regular basis, I primarily use C for embedded applications, but Ruby and Python are very useful languages to have around when I need to do something complex quickly. They will definately help your productivity, not hurt it, so well worth a look.

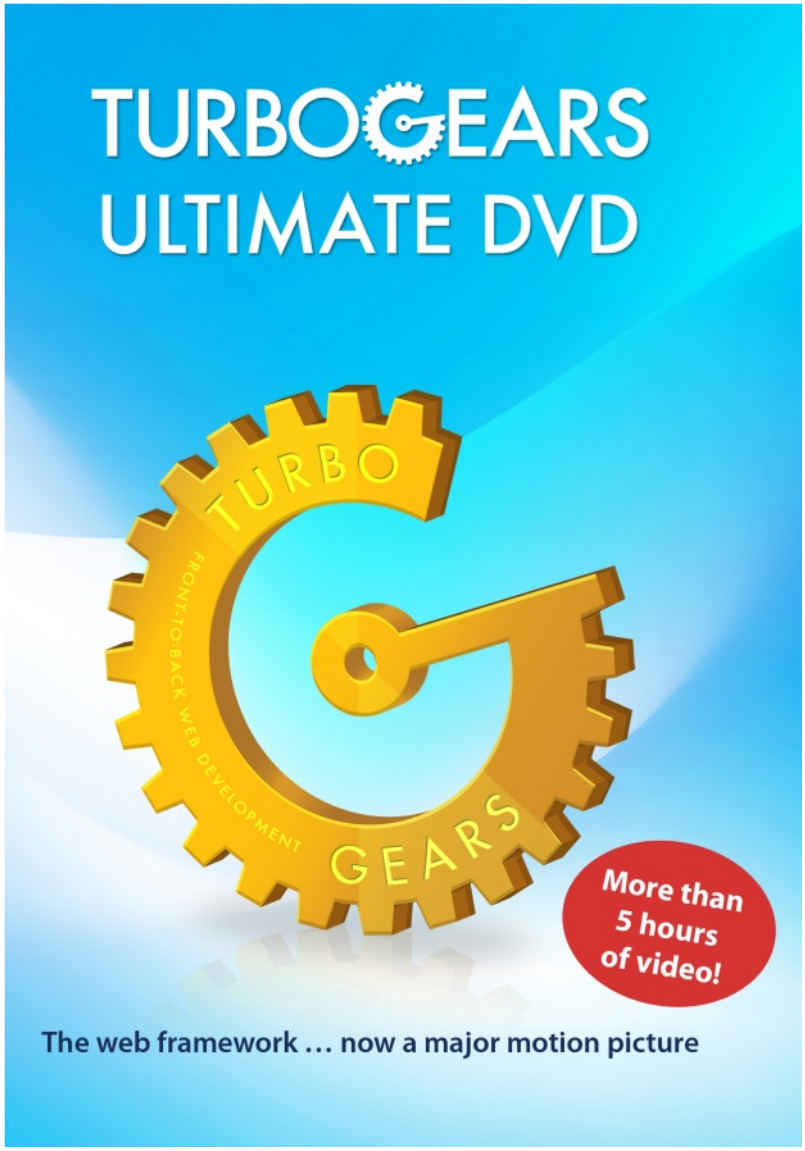
OHIO LinuxFest 2006

COLUMBUS, OHIO | SEPTEMBER 30, 2006 | WWW.OHIOLINUX.ORG

Linux and Open Source Conference and Expo

“A whirlwind tour...”

- Fictitious Daily Newspaper of Great Stature



Learn more about topics covered in this magazine!

TurboGears project leader Kevin Dangoor talks about generic functions and widgets, and takes you on a tour of the TurboGears code. This exclusive DVD-ROM includes high-quality H.264 encoded videos. It's playable on Windows and Mac using the free QuickTime 7 player, or on Linux using VLC, MPlayer or Totem. The DVD also includes an offline browsable copy of the TurboGears, CherryPy, SQLObject, Kid and MochiKit documentation.

Featured Videos

- Using JavaScript with TurboGears
- Introducing Generic Functions (and their use in TurboGears)
- How Widgets Tick (widget package walkthrough)
- TurboGears Core Tour
- The Future of TurboGears
- The 20 Minute Wiki
- Effective Ajax with TurboGears (Flash format of PyCon 2006 talk)

A BLAZING THINGS RELEASE
"TURBOGEARS ULTIMATE DVD" STARRING KEVIN DANGOOR, WITH SPECIAL GUEST BOB IPPOLITO, SOFTWARE CONTRIBUTIONS BY KEVIN DANGOOR, IAN BICKING, REMI DELON, BOB IPPOLITO, RYAN TOMAYKO, DAVID STANEK, ELVELIND GRANDIN, RONALD JARAMILLO, JEFF WATKINS, DAN JACOB, MAX ISCHENKO, SIMON BELAK, ALBERTO VALVERDE GONZÁLEZ, MICHELE CELLA, JORGE GODOY, ONDREJ ZARA, IRMEN DE JONG AND MORE

WRITTEN AND DIRECTED BY KEVIN DANGOOR

Save $5 with coupon code o3mag.

a quick look at turbogears

KEVIN DANGOOR OF TURBOGEARS PROVIDES A QUICK INTRODUCTION TO THE POWER OPEN SOURCE WEB APPLICATION FRAMEWORK FOR PYTHON DEVELOPERS

by Kevin Dangoor

TurboGears is an integrated, front-to-back web framework written in the Python programming language. Others say "full stack", but I like front to back, because TurboGears includes tools to help from the front end (JavaScript running in your user's browser) to the back end (the database). TurboGears integrates a number of different tools that are useful on their own, building pieces on top that really boost your productivity.

Python is a great language for web applications. Python:

- Neatly combines object-oriented, procedural and functional language features.
- Is very readable
- Runs everywhere
- Lends itself well to automated testing
- Is reasonably performant and provides many ways to optimize the hotspots of your application.

The two most common development models on the web today are "code in page" and model view controller (MVC). "Code in page" is enormously popular because it's a natural fit for PHP, and it's a very quick way to get things done when your application is not complex. TurboGears follows the MVC pattern, which lends itself better to larger applications and long term maintainability. MVC also makes it much easier to let designers focus on design and programmers focus on code.

TurboGears was born of necessity. When I started work on my Zesty News product, I had to pick and choose among all of the choices of template languages, web frameworks and database tools in Python. I also needed to document how to work with the tools so that people could write plugins for Zesty News. Having gone through that effort, I thought it would probably help others if I released the whole shebang as an open source package. TurboGears was released in September, 2005 to a huge outpouring of enthusiasm.

Turbo Gears includes:

- SQLObject, an object-relational mapping (ORM) database layer to manage your model objects
- CherryPy, a web framework that handles URL resolution to find your controller code. It also provides a robust server.
- Kid, a template language that is both designer and programmer-friendly.
- MochiKit, a JavaScript library that is well-written, well-documented, well-tested and even a bit Pythonic.
- A supporting cast of several other tools. A notable one is the new setuptools library which provides a useful cross-platform packaging and plugin mechanism for Python libraries.

The first released version of TurboGears didn't have that many features of its own. It was mostly packaging and documentation for the collection of components. It did provide assistance to help developers get up and running quickly. As the project has grown, however, the integrated features that TurboGears provides have grown tremendously.

TurboGears 1.0 builds on top of those components to offer:

- Internationalization
- Widgets
- Identity
- The tg-admin command line tool
- The Toolbox
- Choices of template and database engines
- Task scheduling
- Validation
- Testing tools
- RSS/Atom feeds

easy install

TurboGears takes advantage of Phillip Eby's setuptools, which is destined to be a core part of a future Python version. Once you have setuptools, you have a new command line tool, `easy_install`, that makes installing Python packages trivial, regardless of your platform. It finds the proper download site for the package and downloads it and all of its dependencies with one command. TurboGears would have to be packaged very differently if setuptools did not exist, because `easy_install TurboGears` installs several packages. Installing that many packages by hand would be a pain without setuptools.

Additionally, setuptools provides useful plugin capabilities. As soon as you've `easy_install`ed a plugin, it is available to programs that can use it, without any additional configuration or path setup. TurboGears uses this feature extensively.

tg-admin

One of the first things encountered by a new TurboGears user is the `tg-admin` command line tool. The `tg-admin quickstart` command will get you running with a new project in seconds, giving you a start script to have a running development web server right away.

`tg-admin` has other useful commands, including an internationalization (i18n) helper (see below), database creation tool, a wrapper around Python's shell that adds easy database access, and a command to launch the Toolbox (see below). You can also install new command plugins for `tg-admin` or write your own.

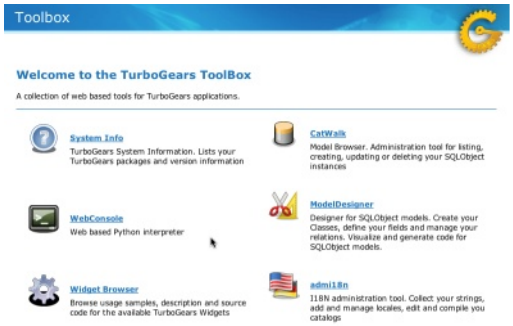
The Toolbox

The TurboGears Toolbox provides a browser-based graphical interface for working with your TurboGears projects. CatWalk lets you browse and update your database, but works in terms of your Python objects rather than the raw database. Model Designer lets you visually design your model classes and generate the Python code at the end. The Widget Browser lets you preview and get additional information about all of the widgets on your system (see below for more on widgets). admi18n helps you manage the i18n aspects of your projects.

Like `tg-admin`, the Toolbox can be extended with new tools. And, the tools are all just TurboGears apps, so they're easy to write!

Internationalization

TurboGears offers internationalization features throughout. TurboGears offers a function for providing locale-specific text to the user at the time of request.



This function is available throughout your Python code and in your Kid templates. There is also a special feature we've added to Kid to allow you to translate sections of your Kid templates with little effort.

As mentioned earlier, TurboGears provides both command line and Toolbox-based tools for managing the i18n process: extracting strings from your code, updating and compiling your translated message catalogs, etc.

As long as you are consistent in your treatment of strings, Python has excellent Unicode support. TurboGears' code is designed to make working with Unicode as transparent as possible, providing consistent encodings of everything that heads out to the web from your application.

Widgets and Validation

TurboGears Widgets wrap up JavaScript, CSS, HTML and image files into easy to use and customize objects. The Widgets package is designed first and foremost to elegantly handle forms. Creating a form from widgets is easy and ensures a consistent appearance wherever you use that form in your application. Additionally, forms make validation easy. You just tell TurboGears that a method is expecting its input to come from a specific form widget, and any validators used by those widgets are verified when the data arrives.

Validators in TurboGears are based on Ian Bicking's FormEncode package. In addition to validation, they provide conversion to and from Python. This means that when you use the `IntValidator`, your Python code is assured of getting an integer when it runs, and not just a string.

If the validator fails, an error handler takes over. For each method in your controller code, you can tell TurboGears which method or function errors should be sent to. For validation errors on a form, you can trivially have the form be redisplayed with error messages for the user.

TurboGears includes many widgets, and you're able to easily look through them via the Toolbox. The widgets vary from simple text fields up to the Ajax-driven AutoCompleteField. The AutoCompleteField is an example of a widget that includes JavaScript, CSS, images and HTML all at once. When you use that widget in a form, all of the necessary resources are automatically included. No need to manually write out script tags or link tags.

Identity

Authentication and authorization needs can vary dramatically from project to project. A very common pattern is the notion of having users, permissions and groups (or roles). TurboGears provides this kind of authentication and authorization right out of the box. Securing parts of your web application could hardly be easier.

Flexible Output

In a typical TurboGears controller method, you return a dictionary of values that gets plugged into a template for output. It looks something like this:

```
@expose(template="o3demo.templates.foo")
def foo(self):
    return dict(current_time = datetime.now(),
                welcome_msg="Hi!")
```

When a user hits /foo, the output is generated by calling the foo method, taking that returned dictionary and plugging it into the template called "foo" that you have in your project. It turns out that this style of output handling has many advantages.

Application testing is very important and TurboGears includes useful tools for testing. The standard testing interface used is a package called Nose, which is compatible with Python's standard unittest module but makes a number of things easier. TurboGears includes a function for running a request from within the testing process without a separate web server. That makes testing running fast. Additionally, due to the unique style of passing data to the templates mentioned above, TurboGears provides a function that lets you examine your controller output *before* it gets to the view. You don't need to parse out the HTML for every test: just look at the values in a dictionary and make sure that they're correct.

The general TurboGears philosophy is to provide one clear path to getting things done. However, it's impossible to anticipate every need, and it's also impossible for the tools included with TurboGears to truly cover every case. One example: Kid is fantastic at generating HTML and XML, but it's not the best template language for generating plain text or CSS. For that reason, TurboGears makes it very easy to use another template language with the same ease that you can use Kid. In fact, Cheetah is included in the download. In the short example above, you could say template="cheetah:o3demo.templates.test" to plug the data into a Cheetah template (without changing the code in your method at all).

It also turns out that returning a dictionary is useful for Ajax. TurboGears can, at your request, turn that dictionary into JavaScript Object Notation (JSON), letting you use the same code to provide HTML to web browsers and a JSON-based API for Ajax or web services.

Other Features

Here are a few other items of note in TurboGears' core code.

TurboGears has a built-in, easy to use task scheduler. Do you have a clean up job to do every hour? It's easy to set that up. And the scheduler can run the job in a separate thread or even a separate process depending on the job's requirements.

You can actually use any Python database technology with TurboGears. Specific support is included for SQLObject and SQLAlchemy, which will cover the vast majority of database needs that applications have. SQLAlchemy is a newer object-relational mapper that is among the most powerful and flexible ORMs available for *any* language. It works equally well with brand new databases *and* legacy databases, which is an area in which many mappers fall short.

TurboGears also provides the built-in ability to generate RSS and Atom feeds, since those formats are common to such a wide variety of applications today.

But wait, there's a lot more!

Those are just the features that TurboGears builds on top of the four core components. MochiKit, Kid, CherryPy and SQLObject all have their own impressive feature lists. See the annotated examples for a feel for each of these packages.

Project Status

The core packages have had stable releases for a long time. TurboGears 0.8 has been in production use by companies and individuals since November 2005. TurboGears 1.0 will be the stable version that comes out from the work on 0.9, which has had alpha test

releases since February 2006. 0.9 is in production use in some places, and the API is stable. As I write this, the main task we're working on for 1.0 is bringing the documentation up to speed and ensuring that the docs remain up-to-date after the 1.0 release.

In addition to the work on the free documentation on the web, there is additional information available via my "TurboGears Ultimate DVD" (http://www.turbogears.org/ultimate.html) and the forthcoming book "Rapid Web Applications with TurboGears" (http://www.turbogearsbook.com/).

Fast, Fun and Maintainable

TurboGears provides a fast and fun way to write applications that are maintainable and high-quality. Your customers will be happy with how quickly new features are implemented, and you'll be happy with the fact that you could focus on the application at hand rather than the mundane details of HTTP parameter marshaling or XML-based configuration files that plague other frameworks.

The TurboGears focus on simple APIs to get you going and a smooth learning curve as your needs increase is sure to make you happy you checked it out.

About the Author

Kevin Dangoor is project lead for TurboGears and founder of Blazing Things LLC. You can keep up to date on all things TurboGears by checking out Kevin's blog over at http://www.blueskyonmars.com.

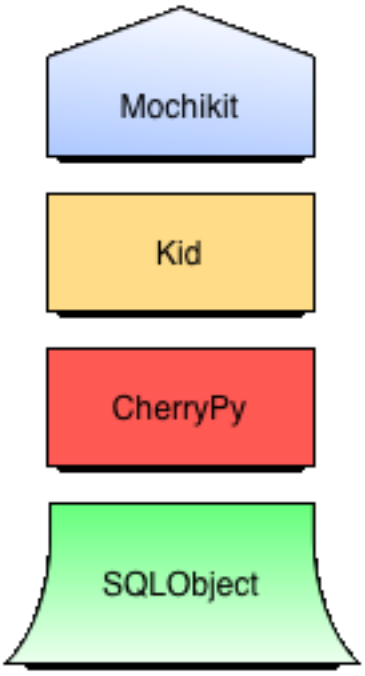


Figure 20.1 - Third Party Components

TurboGears takes the best components available and combines them into one easy-to-install, documented whole. TurboGears includes parts that join the pieces together and make them work together seamlessly, but doesn't obscure each included project. This allows you to take advantage of all existing documentation, articles, mailing lists and other resources that have built up in the communities for each project.

From frontend to backend:

- MochiKit is a clean and powerful JavaScript library
- Kid is a designer- and programmer-friendly template system
- CherryPy makes doing web input/output as easy as writing a Python function!
- SQLObject lets you access your database as you would normal Python classes, without obscuring the database itself.

Anatomy of a TurboGears Project

When you run tg-admin quickstart, this is what you get

TG's configuration is easy to use, but you often don't even need to touch these files. These bits of config go wherever you deploy your app.

CherryPy hands off web requests to your code in here. The stuff in this file figures out what to do with a request and which template to display.

You can customize how JSON output is created.

You can define yuor database in Python code here, or have the Python classes look in the database for their definitions.

Testing is important, and TG helps you get started by including sample tests in your new project.

```
o3demo/__init__.py
o3demo/config
o3demo/config/__init__.py
o3demo/config/app.cfg
o3demo/config/log.cfg
o3demo/controllers.py
o3demo/json.py
o3demo/model.py
o3demo/static
o3demo/templates
o3demo/templates/__init__.py
o3demo/templates/login.kid
o3demo/templates/master.kid
o3demo/templates/welcome.kid
o3demo/tests
o3demo/tests/__init__.py
o3demo/tests/test_controllers.py
o3demo/tests/test_model.py
o3demo.egg-info
README.txt
dev.cfg
sample-prod.cfg
setup.py
start-o3demo.py
```

master.kid provides your app-wide navigation. login.kid is customizable if you want to change the login screen.

setuptools uses this egg-info directory to keep track of metadata for your project. This is how you tell setuptools that you've created an amazing new Toolbox tool or Widget.

In addition to the application config in the config directory, you have deployment-specific configs. You'll generally just need to tell TG where your database is.

Your new project includes a Python-standard setup.py script. You can get your new project distributed in the Python Cheeseshop with no extra work. The Python Eggs created by setup.py are also great for deploying your app.

There's nothing like instant gratification! You can fire up a web server immediately by running the start script! When in development, the server automatically restarts as needed when you make changes.

A Look At A Controller

The default method responds to anything that the controller receives that doesn't match an existing method. In this case, going to /HiThere will call the index method with the name of HiThere.

The imports were removed for brevity. TurboGears keeps things explicit: you know where a piece of code you're using is located, because you imported it.

Here is a form with some required text fields and a select box. The names of these fields correspond to the names in the model class, which makes using this form very easy.

```
class AddressFields(WidgetsList):
    name = TextField(validator=String(not_empty=True))
    address = TextField(validator=String(not_empty=True))
    citystate = TextField(label="City, State",
                          validator=String(not_empty=True))
    country = SingleSelectField(options=
          [("usa", "USA"), ("can", "Canada"), ("mex", "Mexico")])

addressform = TableForm(fields=AddressFields())

class Root(controllers.RootController):
    @expose()
    def default(self, name):
        return self.index(name)

    @expose(template="o3demo.templates.welcome")
    @identity.require(identity.not_anonymous())
    def index(self, name=None):
        if name:
            address = Addresses.byName(name)
        else:
            address = None
        return dict(address=address, form=addressform)
```

@expose tells TurboGears to make a method available on the web. The returned dictionary is passed in to the template to generate HTML. The index method is the "root" of the site.

This call to @identity.require ensures that you have to be logged in to access this information.

Addresses is our SQLObject model class. We're doing a database lookup by the person's name.

We return the appropriate model object and the form so that the template can display the information. If it made sense to do so, we could use more expose decorators to produce different output formats.

A Look At A Controller Part 2

There's no template listed here, because we're not producing output from this method. We're redirecting instead.

This @validate tells TG that we're expecting the fields from this form to come in. This will check our required fields for us.

If the user's input fails validation, this error_handler declaration says to return them to the index method. Because of the way TG widgets work, the bad data is redisplayed with error messages without any additional coding.

We've specified that you have to be in the "admin" group to use this method. The user can view the form with the index method, but to actually save requires this group.

```
@expose()
@validate(addressform)
@error_handler(index)
@identity.require(identity.in_group("admin"))
def save(self, **kw):
    try:
        address = Addresses.byName(kw["name"])
        address.set(**kw)
        flash("Changes saved!")
    except SQLObjectNotFound:
        address = Addresses(**kw)
        flash("New address added!")
    redirect("/" + kw["name"])
```

The save method is where we tell our form widget to post to.

We look up the person's name in the database. If they're there, we change their record with the set() method. Otherwise, we INSERT a new row in the database by creating a new Addresses object.

flash() will display a new message at the next possible opportunity. In this case, we're doing a redirection, so the flash message will actually be displayed on the next request... as a TG user, you don't need to think about that, though!

Incoming request parameters automatically become Python method parameters. We could have named them all, but bringing them in as a dictionary using the Python standard **kw was easier.

(IN)SECURE

Open. Informative. To the point. (IN)SECURE Magazine is a free digital security magazine discussing some of the hottest information security topics.

// www.insecuremag.com //

turbogears deployment and scaling

JEFF MARSHALL AND GREG LIN OF FROZENBEAR INC LOOK AT THE DEPLOYMENT OF TURBOGEARS AND EXAMINE HOW WELL THE FRAMEWORK SCALES TO MEET THE DEMANDS OF A GROWING NETWORK

by Jeff Marshall and Greg Lin (http://frozenbear.com)

Motivation

At FrozenBear we have been rolling out websites with TurboGears since 0.8. We wanted a Python framework that was easy to use, deploy, and scale and TurboGears has fit the requirements. This article will discuss our experiences with “easy” deployment and scaling. Know what a basic TurboGears project looks like before reading on.

Deployment

Deploying a TurboGears (TG) project is easy.

1) Build a python egg
2) easy_install the egg on the production server.
For an imaginary TG project named “foo”:

```
#
# make the egg
#
# [on the dev machine]

cd /projects/foo
python setup.py bdist_egg
```

The resulting egg shows up in /projects/foo/dist/foo-1.0-py2.4.egg.

```
#
# deploy egg
#
# [on the dev machine]
cd /projects/foo/dist
scp foo-1.0-py2.4.egg \
 user@production:/var/www/foo/eggs

scp /projects/foo/sample-prod.cfg \
 user@production:/var/www/foo/prod.cfg

# ...edit your prod.cfg as needed...

# [on the production machine]
cd /var/www/foo/eggs
easy_install foo-1.0-py2.4.egg

# [on the production machine]
cd /var/www/foo
mysqladmin -h localhost -u root -p create foodb
tg-admin sql create --egg foo

# run the server
/usr/bin/foo-start.py
```

By default, the easy_install command creates a startup script in /usr/bin/foo-start.py and puts the egg's files in the directory /usr/lib/python2.4/site-packages/foo-1.0-py2.4.egg. tg-admin creates the tables for foodb and foo-start.py launches your site.

Scaling

Now that the site is running on the production machine, think about how it will handle growing traffic.

The ability of the site to scale will depend on the various layers between the TG server and the outside world as well as the efficiency of the communication between the TG server and its database. Develop a configuration appropriate for your plans. Think about which parts of the system will become performance bottlenecks and the simplest ways those bottlenecks can be fixed. Scaling usually requires a balance between adding hardware and optimizing code. Generally it is cheaper and more time-effective to ensure the application will scale with simple hardware addition rather than relying on future code optimization efforts. Time spent on application over-optimization will almost always cost more than adding extra machines. If your TG server is getting loaded down by traffic, get more machines to run another instance of the TG server and let a load balancer spread out the requests. When the database requests grow too big, partition/replicate/cluster the database instance across more machines. The project needs to be designed with these scenarios in mind to make scaling easy.

Components we've found helpful for those tasks are Apache, Squid, and Pound:

- Apache can be placed in front of TurboGears to serve the content using HTTP 1.1 (the CherryPy server uses HTTP 1.0 at the moment), to more efficiently serve up static content, and to handle SSL

certificates.

- Squid can be used as an HTTP accelerator by caching content. CherryPy has a built-in caching filter, but Squid can provide much more powerful caching options while removing load from your CherryPy server.

- Pound is an easy-to-use software load balancer. By all means, move on up to hardware load balancers if your project can afford it. But Pound can pull a lot of weight for you in the meantime.

To demonstrate these components and how they can help scale, we will walk through a case study of a TurboGears-based web service FrozenBear has recently built called MeCommerce (http://mecommerce.goodstorm.com/). MeCommerce is a useful case study because we needed to incorporate each of the above components in various ways to successfully scale.

There are two primary TurboGears processes running this system: a secure order form and payment processing server running behind a SSL, and a non-secure server handling a product box showing in iframes on many blogs and websites along with a publisher website to manage accounts. It was easy to identify that the product box would have the most scaling issues since the iframe needs to get fresh content every few minutes for each blog it sits on, and at the same time it needs serious caching because of the heavy traffic it experiences. We need to be able to add more product serving capacity very quickly and also add more instances our of TurboGears servers if needed.

Pound

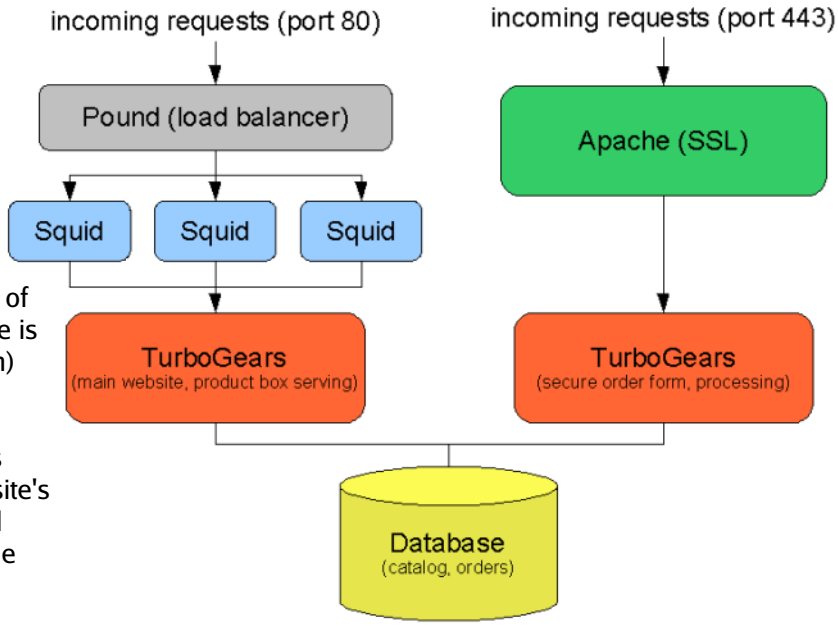
Pound is available for download from http://www.apsis.ch/pound/. Pound handles the incoming (non-secure) requests and balances the load to the bank of Squid servers. (We'll deploy Pound at the front of the secure request route when the secure traffic demands multiple instances of the TurboGears order processing server. The secure is intentionally kept separate from the non-secure path) The load balancer automatically stops forwarding requests to servers that are down, and new Squid servers can be added as-needed when traffic levels outgrow the existing capacity. Squid will improve a site's overall uptime and responsiveness. A simple Pound (version 1.9) configuration file such as this will do the job:

```
User daemon
Group daemon
ExtendedHTTP 0
WebDAV 0
LogLevel 1
Alive 30

# your server's IP address and port where pound
# listens

ListenHTTP xxx.xxx.xxx.xxx,80
UrlGroup ".*"

# each of your squid servers' IP addresses and ports

BackEnd xxx.xxx.xxx.xxx,80,1
BackEnd xxx.xxx.xxx.xxx,80,1
BackEnd xxx.xxx.xxx.xxx,80,1
BackEnd xxx.xxx.xxx.xxx,80,1
EndGroup
```

Pound can be configured to forward specific, uncacheable paths directly to the TurboGears server and bypass the Squid pool.

Squid

The Squid Caching Server can be obtained from http://www.squid-cache.org/. The Squid servers are configured as HTTP accelerators by setting the httpd_accel options:



```
# the server's hostname
httpd_accel_host myinternal.domain.com

# the TurboGears port
httpd_accel_port 8001
```

Squid will automatically cache the TG server's static content. This is one of the most important steps you should take when preparing your project for production traffic. CherryPy does a fine job of serving up static content, but we highly recommend that you use Apache or Squid to serve your static content instead. Not only will Apache and Squid do a better job of serving up static content, they will take the load off the CherryPy server so it can focus on the more import tasks it needs to do. We use Squid to serve static content on the non-secure requests, and we take advantage of Apache's mod_rewrite to serve static content on the secure request path.

Squid can help cache more than the static content. For MeCommerce, it was critical that Squid handle the bulk of the product box iframes. The product box iframes appear on many blogs and websites, far eclipsing the other traffic that MeCommerce serves. By default, Squid will cache content from the TurboGears server according to the HTTP caching headers that it sends back. To set the caching headers, create a CherryPy filter:

```
import cherrypy
from time import strftime, gmtime, time

class
 ExpiresFilter(cherrypy.lib.filter.basefilter.BaseFilter):

   def beforeFinalize(self):
      if not cherrypy.config.get('expiresfilter.on', False):
         return

      cache_seconds =
        cherrypy.config.get('expiresfilter.seconds', 300)

      cherrypy.response.headerMap['Last-Modified'] =
        strftime('%a, %d %b %Y %H:%M:%S GMT',
        gmtime(time()))

      cherrypy.response.headerMap['Expires'] =
        strftime('%a, %d %b %Y %H:%M:%S GMT',
        gmtime(time() + cache_seconds))
```

Add this filter to the CherryPy root before starting the server:

```
cherrypy.root = Root()
cherrypy.root._cpFilterList = [ExpiresFilter()]
cherrypy.server.start()
```

Now, set "expiresfilter.on = True" in the config file for each path Squid needs to cache. The "expiresfilter.seconds" option sets the expiration time for each path.

Apache

Apache is available from http://httpd.apache.org/. In the MeCommerce example, Apache is used primarily to handle the SSL certificate on incoming requests on port 443. However, we can also take advantage of Apache to handle static content along that path. Using mod_rewrite, configure Apache like this:

```
<VirtualHost _default_:443>
# ... your standard ServerName and SSLEngine config
# lines...
# use mod_rewrite to route requests
RewriteEngine on
# use mod_rewrite to serve up the static content
# (set the path to your static content; it is a good idea
# to create a symbolic link to your deployed egg's
# static path.
# unfortunately you will need to update this symbolic
# link each
# time you deploy a new egg version, but it's easier
# than updating
# this Apache config each time)
RewriteRule ^/static/(.*) /path/to/static/$1
# send the rest of the traffic to your TurboGears server
# (set your domain and TurboGears port)
RewriteRule ^/(.*)
http://myinternal2.domain.com:8002/$1 [P]
</VirtualHost>
```

Definitely have Apache handle static content if static files are larger than 100KB (i.e. images).

Conclusion

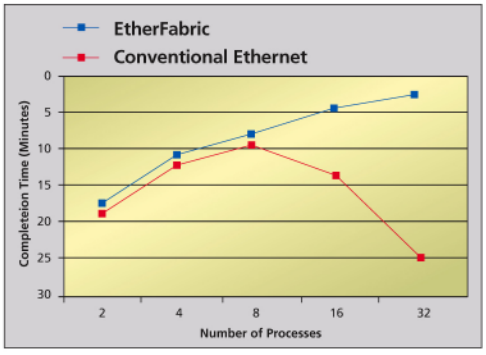
No framework can guarantee scalability, but a good framework like TurboGears gives you the latitude to make a variety of scaling decisions. Be smart in the design. Code as if the project will be initially deployed on two or more servers. Your specific use case will dictate which pieces need more attention. Don't overly optimize or hardware-scale until you know the site's usage profile. We highly recommend TurboGears.

ACCELERATE APPLICATION PERFORMANCE!

>> HALF THE LATENCY

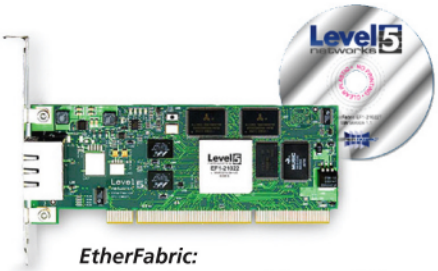
>> TWICE THE BANDWIDTH

>> 4X THE PERFORMANCE



Take EtherFabric for a ride today and experience the accelerated performance for yourself.

Visit **www.level5networks.com/landing/3.php** and take advantage of our limited time offer to ship you one extra EtherFabric NIC with your initial order.



EtherFabric:
High Performance Ethernet NIC

Level 5 networks

Level 5 Networks Inc. • 840 West California Avenue, Suite 240 • Sunnyvale, California 94086 • 408.245.9300

unsecure wifi: the outside threat from inside

WIRELESS NETWORKS ARE AN IMPORTANT PART OF IP MOBILITY SOLUTIONS, DEPLOYING SECURE SOLUTIONS CAN BE A CHALLENGE. THIS ARTICLE LOOKS WIRELESS SECURITY FROM THE MALICIOUS USERS PERSPECTIVE

by John Buswell

The usual WiFi security article would discuss the various ways to protect your network, WPA2, disabling SSID broadcast, firewalling your WiFi network as if it was part of the Internet, and so on. This however, is not your usual WiFi security article. Unsecure WiFi poses a serious security risk to any campus, business, or ISP network; this article uses data from real world examples to show how easy it can be for a malicious user to cause serious problems. Sometimes its useful to look at security from the other side. In this article we take a unique approach, looking at things as if we were a malicious user, what would we do and how would we do it? No wireless networks were harmed during the research for this article.

Reasoning

There are plenty of reasons why a malicious user might want to access a network illegally. On one hand you have people who just like to cause trouble. You also have people who want to make money through spam, botnets, and so on. On a more practical level, you have people who want to collect credit card information, information for identity theft, and people who want to collect logins to a larger network. You have people who want to do something illegal on the Internet and not get caught. It could be illegal content such as child pornography, terrorist communication or cyber terrorism such as distributed denial of service attacks.

Unsuspecting Victim

The unsuspecting victim here is the person or organization who had what they thought was a secure solution deployed, but were then blamed for some illegal activity that they were unaware of. In the case of identity theft, imagine a careful person who shreds documents with sensitive information, destroys old credit cards with a media shredder, uses Firefox, pays for security tools, and runs a firewall behind their cable modem. They don't have to be tech savvy, but they've researched enough to be secure. When this person goes to buy a new PC, they pick up a laptop, it has WiFi, and the sales person sells them a WiFi access point or router. It plugs in, works great, and they can use their laptop without wires from the sofa, the kitchen and even bed. Is it secured? Do they know? Do they care? Probably not.

Finding suitable victims

The best location for finding victims are where people are clustered. So apartment complexes, townhouse complexes, and college dorms are good starting points. Office districts and industrial estates also offer pretty good targets depending on the town or city, and the technical capabilities of local companies.

For the purpose of this article, we're going to look at a real world situation. Since this applies to practically any town or city where broadband Internet is available, we're going to use data combined from a number of undisclosed college towns. These are nice, and peaceful towns, whose citizens are unaware that they could easily become an unwitting victim of some thug with a laptop and some wifi gear. Like many college towns the summer it is pretty quiet; when the students are in town in the fall (autumn), it can be extremely busy. Most college towns have colleges that offer a wide selection of IT and non-IT courses. There are a handful of IT companies in town, and a couple of ISPs. The mileage you'll get with the local IT businesses varies; some are nothing more than glorified PC repair and office supply companies who are run by business people turned IT professionals. Learning to install Windows Server 2003 and Exchange over the period of a few months doesn't classify you as someone who should start selling IT solutions, but unfortunately that doesn't stop people. A little knowledge is a dangerous thing, and unfortunately for non-technical people, someone who appears to know a lot more than they do must be an expert. Expertise is relative, and unfortunately many business owners don't take this into account when purchasing IT solutions.

Equipment

I wanted to keep this article relatively simple. So although I could point to great open source tools that you can install under Linux to do wireless packet sniffing, scanning and so on, I'm going to stick with off the shelf things that you can acquire at your typical national retail store. If you are not from the United States, there are plenty of stores that resemble a huge warehouse that sells practically everything for

reasonably low prices, and there is some form of these stores in or near practically every town in the United States.

Lets take the scenario that the malicious user is out of town. Most car rental companies offer unlimited mileage, and most will rent you a nice minivan or SUV for under $40.00 a day. The minivan is a good option, as it is less suspicious, most have tinted windows, and you can store equipment such as UPS, full sized servers and so on in the back undetected. You can pick up vans from airports too, so theoretically a terrorist could easily fly into any country, with cash or credit cards on them and nothing else.

On a quiet Saturday evening, I grabbed a shopping cart (“trolley” for the Europeans) and took a stroll around one of these large stores to see how much it would cost. Now I didn't buy any gear since I already had it, but I did ask the some store representatives how easy it would be to return; even high value items such as laptops if I didn't like them. It turned out to be a lot easier that I expected. So technically someone could fly in, rent a car, buy gear at one of these warehouse stores, take care of business, return the gear to the store, return the rental car and fly home with minimal costs!!

So what do we need? The first thing on our shopping list is a power inverter. A power inverter is a small metal box that you plug into the DC powered cigarette lighter socket on your car, and gives you AC power outlets so you can run regular AC devices, such as a laptop power charger in your car. Local store - $17.69. Next on the list was an APC power strip, we don't want to fry that new laptop, so again the local warehouse store, a few aisles over, for $14.95. Now a nice laptop (again at the local store) with built-in WiFi and running Windows XP -- $697.00. Finally, our malicious user is smart- they want to make it harder for authorities to investigate should they get caught- so they buy a few USB WiFi adapters and PC Card adapters, so not everything comes from the same MAC address. USB WiFi adapters at both warehouse and office supply stores - $39.95 each, PC Cards ranged from the same price up to $69.95. Interesting enough, the local warehouse store had absolutely no 802.11a equipment, not even 802.11a/b/g routers. Everything was 802.11g.

Now if we wanted to pickup some WiFi routers to duplicate victim's routers, perhaps to packet sniff by bridging between their real access point and our fake one, then the local warehouse store had everything we needed including 802.11g range extenders and 802.11g access points.

Our really smart attacker, makes a note of all the brands and models of access points on sale at the local warehouse store. Chances are, our unsuspecting victims bought their equipment here. In our college town USA, the only other place to buy gear is a national-chain office supply store. Thankfully, they were well-stocked with 802.11a capable equipment, so those users aren't safe either.

So, for about $900.00 we could be well-equipped, courtesy of just one local store. As I said, we didn't buy any of the gear as we already had it, however it's very easy to do as our research has proven.

False Sense of Security

While hanging around the WiFi aisle at the local warehouse store, I ran into a number of people looking at WiFi, sometimes eagerly recommending WiFi solutions they had previously bought. Most stores stock Belkin, Linksys, and some off brand WiFi. One student told me to buy Linksys because it had Internet Security and thats all you needed. He was referring to the logo of Symantec Internet Security on the box that provides an Anti-Virus and Firewall in a 60-day trial version.

Now Linksys has gone to some effort to make it easier to configure WPA2 through their “Secure” Easy Button. Unfortunately, this is just something new, and most students already purchased WiFi equipment. Many laptops don't support the Secure Easy Setup, and $69.95 for the external PC card is just a waste when you can buy books, beer, or pay rent instead, and besides, your laptop has WiFi.

What's worse is that companies looking to off-load old stock are selling 802.11b equipment for $9.95. A nice discount basket at the local office supply store, and for under $40.00 you can build a small 802.11b network. To students, $9.95 looks a lot better than $70 and it's all 2.4GHz wireless anyway, right?!

Getting Software

A number of local establishments offer FREE wireless Internet to their customers, one ISP offers FREE wireless Internet in select downtown areas. Duplicating our malicious user, we drive to one of these FREE areas, park our vehicle and grab the laptop. A quick coffee or two later, and our laptop is now equipped with Netstumbler, Firefox , and some other tools. We also had some time to poke around with traceroute and ping to get some inside information on the ISP. Again, all perfectly legal.

Various local websites and google maps provide us with enough information to plan our route, and show us where the student dorms and off campus clustered

housing is located. Finally, we quickly download the manuals from the various support sites for the routers and access points we saw on sale at the local stores. Now we have the default username and passwords. A really clever attacker would also jot down the first couple of octets from the mac addresses, clearly printed on the packaging, just in case Netstumbler can't identify the brand from the mac address during the scans. Back to the car, and it is time to look around.

Important Legal Notice

At no point were any access points accessed or modified during the research for this article. It is important to keep that in mind, as this article discusses what an attacker could have done.

The Results

Our new town isn't a large town, so an hour long drive at just under the speed limit around town, revealed 781 access points in the area. About half of the access points had WEP enabled. About 40% had default SSIDs, but some had WEP enabled, we worked out that roughly 30% of the visible access points on our route, which covered just the main streets, had no WEP and default SSIDs. While we didn't connect to any devices, chances are the majority of them probably were configured with just the default admin passwords.

It was pretty easy to see a pattern between ISPs. For example, one DSL provider had serialized SSID numbers, I know this because I happened to use a small cafe that offered FREE wifi using a specific DSL provider. A quick hit to any IP lookup page by a hacker over one of those serialized SSID connections which weren't secured, would make it easy to determine which had DSL and those that had something else. The serialized SSIDs are a serious problem, painting a target on the customers WiFi network, and likely one of those self-install kits.

I was glad to see that one ISP was rolling out equipment to their corporate customers with secure, commercial solutions instead of trying to sell off-brand or retail gear. A couple of hotels looked like they were either wide open or possibly used MAC address filtering, thats just a guess, giving credit to the ISP who rolled it out.

Many of the broadcast SSIDs revealed addresses, names or organizations. With most of the channel usage centered on channels 1, 6 and 11, with 6 being extremely popular and the default setting with Linksys. It almost look like the SSID was being used to either make sure they didn't hop on the wrong unsecured WiFi or as a warning to other people hitting their AP by mistake.

Now we drove around on a Sunday morning, I would imagine that many more access points would have shown up had we drove around during a school night, as some people do turn off their access points when they are not at home.

The Outside Threat from Inside

Administrators go to great lengths to prevent bad traffic reaching their users. This ranges from blocking netbios traffic, running intrusion detection and prevent systems, rate limiting and other techniques even on internal switches and routers. Some ISPs even protect customers and users from each other, blocking various ports on the provider side connected to the CPE, thus preventing your neighbor from hacking you or giving you that Internet Worm that they downloaded via IM!

The problem is that all this protection goes out the window when the attacker can simply hit the user's local system over the unsecured WiFi. Often these machines are artificially secure, users relying on the security provided by routers / firewalls instead of updates and software firewalls. Students who often like to play games over their LAN with friends, often disable software firewalls because they block the game traffic from the local LAN.

These lapses in security make it easy for an outsider to install key stroke loggers, remote control software, and other software on the victims computer. Now that person is a risk, a legitimate user, perhaps a VPN user, now has their account information compromised. The malicious user can now gain legitimate access to your network, lurk around, and look for vulnerabilities. At the very least they can read internal email, send email, and perhaps even gain customer information.

What works for that English language student with their WiFi router, also stands true for the sales representative living in an apartment complex. Same level of technical savvy, same level of false sense of security. Same internal threat.

Internal Bot Nets

Now when most providers discover bot nets, typically through bandwidth spikes, security / abuse complaints, they shut down the control channel. What happens, though, when the control channel is outside of their network, run over WiFi by chaining access points together, creating one large network? Remember, most of these unsecure WiFi routers and access points have default SSIDs. So it's not hard for some malicious user to go in, change the config, and make adjacent apartments run on the same network. The attacker may

have secured these access points so nobody can identify them easily, furthermore, they might use a range extender to bridge a completely different access point with their bot net, and use a completely different ISP to access the control channel for the bot net remotely.

If you think thats far fetched, we spoke with a local apartment complex manager who was nice enough to show us around their laundry facilities. It would be easy to stash a Linksys WRT54G running OpenWRT, or a range extender in any apartment complex laundry room. The equipment could sit there for months behind a washer or dryer before anyone would notice. Simply placing a sticker with the logo of a local ISP is simple enough to keep maintenance and management from even questioning the device.

Luxury War-Sitting

Thanks to urban planning, many upmarket college off-campus complexes are located near or next to hotels. Simply book into a hotel, depending on the distance it might be necessary to place a range extender between the complex and the hotel. Here in our example town, at least one hotel was close enough to a complex to scan directly from the hotel. Over a couple of days a malicious user could collect multiple accounts for a wide variety of locations. In a college town like our sample town, it would be relatively easy to obtain account information and thus become a threat for the local colleges and Universities. Thanks to the hotel, it can be done in style, while watching TV, and without running the risk of getting interrupted by curious third parties or the police.

Internal Denial of Service

The attacker doesn't have to have a reason why, perhaps another attacker took over their bot net, or they simply don't like the ISP, or just have a mischievous nature. An internal DoS attack might even be difficult to detect, a number of local customers with high bandwidth connections, downloading files from the same site, something large perhaps some DVD images, looking like normal traffic. A large enough bot net could be used to congest the local ISP network. It could be as easy as bouncing data between local users in a loop, flooding the local network.

Depending on the ISP, this is often easier to do than you might expect. Some providers, especially those that use wireless technologies such as 900MHz, have limited bandwidth between CPE and their towers. They're “betting the farm” that their users won't all consume the bandwidth at the same time; a well-orchestrated attack could look like normal network traffic and cripple the ISP.

But It's Just Timmy's PC

Remember that old Windows 95 box that was the family PC a few years back? Now it sits in Timmy's room. Timmy is now old enough to use the Internet, so dear old Dad goes and buys some WiFi gear because it's cheaper than running cables and drilling holes in the wall. That crimper thing looks strange, since good old Dad works in sales. Now while Dad's laptop is secured, Timmy's is not, and security is only as good as the weakest link on the network. That Windows 95 box is just an invite for an attacker, probably not updated, and what does Timmy need with anti-virus and firewalls, he is just a kid, how much of a threat is CartoonNetwork.com ?

Fixing the Problem

Not everything security related can be fixed entirely with technology. The sooner you realize that, the faster you can go about securing your network.

Education is the first line of defense, and this can be as simple as regular IT awareness sessions that students (or employees) must attend to keep network access. Offering secure WiFi routers, or having IT employees reconfigure home WiFi routers for your non-tech employees, can help improve security to a degree.

In a city-wide problem, especially in a University town, then going through a central organization such as the Chamber of Commerce or the University, will help raise awareness and provide a community solution. This might involve an ISP offering free workshops securing WiFi routers and giving information on how to download free anti virus and so forth. Such events can help attract new customers, and at the very least reduce the threat to your network.

It is not a problem that is going to get resolved overnight, most students carry laptops these days, so for Universities it might just pay off to offer services that check and audit student laptops for security problems – viruses, worms, outdated vulnerable code, keystroke loggers, remote control software and so on.

Remote users who access the network via VPN, could use a simple web based application that forces them to register their location. If you know Dave in Sales has cable at home, and suddenly starts trying to login from a DSL connection, if he is registered for his home location, you can lock out the account if the authentication information matches. If Dave is on the road, and you know he is traveling to a particular city,

have them register that location once they login. Simple web applications that communicate with the firewall through writing or updating data in a database or via LDAP, can provide an extra layer of security.

For example, Dave gets to his hotel, and visits https://vpnreg.mycompany. Before he left the office, he registered that he was going to Boise, Idaho for 3 days. When he hits the vpnreg, he authenticates, then it asks him a series of questions related to his trip and location. If he gets all of the questions right, the location is registered for the days he is away. While he is away, he might only need email and access to presentation documents, this type of location based access restrictions can limit the damage done if his laptop is stolen while at the hotel.

Conclusion

Hopefully this article has given you a new perspective on the problems involved with unwitting customers and users connecting equipment to high bandwidth networks. Security is only as good as its weakest link. While the risk of being fired or expelled will prevent internal users from doing bad things on your network, it won't stop them from doing irresponsible things that can result in third parties gaining their privileges on your network. A combination of education and “thinking outside of the box” is necessary to limit the risk of the outside threat from inside.

About the author

John Buswell is Editor in Chief of this magazine, he is also Chief Technology Officer and co-founder of Spliced Networks, a privately owned Linux appliance company. He will be giving a presentation at Ohio LinuxFest on September 30, 2006 titled "Open Source Zero Day Attack Protection". For further information visit http://www.splicednetworks.com and http://www.ohiolinux.org.

Mandriva

Businesses need rock-solid IT solutions

**Mandriva Linux Corporate Server & Corporate Desktop offer outstanding robustness, scalabilty, and reliability.
All with the ease of use specific to Mandriva products.**

- Full IT solution for server and desktop deployments
- Open standards
- Both x86-32 and x86-64 architectures are supported
- 5-year product maintenance
- 24/7 support
- Mandriva Online update service - Professional Level
- Incredible price

http://www.mandriva.com/business/corporate-server

http://www.mandriva.com/business/corporate-desktop

www.mandriva.com

ata over ethernet

MUHAMMAD HAMMAD LOOKS AT THE OPEN SOURCE STORAGE SOLUTION THAT USES REGULAR LAYER 2 SWITCHES
AOE IS A SOLUTION PIONEERED BY CORAID

by Muhammad Hammad

Data storage space always reaches to its limits, either today or tomorrow. Sometimes there are video/audio data that require huge space or may be a production server generating massive logs or it could be that backup is required on permanent basis. Whatever the reason, one will eventually run out of space and is bound to increase storage capacity. Fortunately hard disk storage capacity is increasing day by day with a steady decline in cost. Over a network, the storage capacity is limited by the machine's hardware i.e. you can only attach limited number of hard disks to a machine. If, somehow, we could just attach an unlimited number of hard disks to a machine(s), we could get unlimited storage capacity. This is what ATA over Ethernet (AoE) protocol supports- i.e. ATA storage devices accessible over Ethernet to other machines in the network.

AoE is an open standard Ethernet based Storage Area Network (SAN). SANs over Fibre Channel are more complex and expensive. Another variation of SAN is iSCSI, which uses SCSI command set over TCP/IP, typically over Ethernet, and is accessible over LAN and WAN. On the other hand, AoE runs on top of Ethernet, which is much cheaper, easy to configure and deploy. Also, AoE is a lightweight protocol since it relies on Ethernet and does not employ the complexity of TCP/IP protocol.

How it works?

AoE works by providing the host servers (web servers, mail servers etc.) access to disk drives through AoE servers. An AoE server is a small computer that has a processor and disk drive on a small printed circuit board. These boards are known as blades. The host sends request messages to the AoE server, which in turn provides block access to the disk, and reply messages are returned back to the host. Each message in AoE protocol communication should be considered an unreliable message. Reliability is achieved by the client host AoE software that resends the request message if no response in a specific time. AoE packets are encapsulated in standard Ethernet frames and share the Ethernet header, and so the AoE servers and other machines can be on the same LAN. However, it is recommended that you separate storage network from other network traffic to achieve the highest performance.

The location of an AoE server blade is identified by a shelf number and slot number. A shelf is composed of a number of slots, and each blade can be inserted into a slot. AoE packet header contains information about the shelf and the specific slot number to access a specific blade. In AoE specification, these numbers are referred as major and minor numbers. AoE devices communicate over Ethernet, based on Ethernet addresses, and thus does not require IP configuration. In order to find an Ethernet address of a special blade, a broadcast message would be sent with the shelf and slot number and that particular blade would respond containing the Ethernet address. The protocol also allows the flexibility to communicate with multiple shelves and blades. For this purpose, a specific broadcast value, all ones, is used for shelf and slot number. For instance, if a shelf value is all ones and slot number is specific then the message will be processed by the specific blade, identified by slot number, of all shelves. On the other hand, if slot number is all ones with a specific shelf value then all the blades within that shelf would respond.

AoE messages

AoE protocol consists of sending messages to the servers and getting reply from the servers. There are two types of messages that are generated- ATA commands and Query Config information. Both these formats have their own fields in a message but they share a common header. ATA commands are request/response messages to perform ATA transactions on an ATA device. ATA transactions perform read/write from the disk. Query Config information is used by the hosts to identify blades. It is more flexible than shelf and slot number, which is not very scalable for a large number of client hosts and AoE servers. Query Config information allows client hosts to set/retrieve special information on AoE servers, which can be used later for identification purposes.

How to implement AoE based SAN in Linux?

AoE is implemented as block device driver in the host operating system to provide the host access to the

storage area network. The driver is responsible for connecting the host to target AoE disks using host NIC. A target disk is a disk that is to be accessed by other machines over AoE protocol. This allows seamlesss integration of host and storage area network, which appears as local block device(s) in the host operating system. AoE drivers are available for Linux kernels 2.4 and 2.6 and FreeBSD kernels 4.11 and 5.3 from http://www.coraid.com/support/index.html. The current Linux kernel version, 2.6.15.6 includes AoE driver, which is not enabled by default. Moreover, download aoetools from http://sourceforge.net/projects/aoetools/. This tool set supports Linux kernel AoE driver and provides useful programs that runs in user space.

Coraid, the original designer of AoE protocol, has also made AoE hard disk known as EtherDrive. In order to implement such a storage area network based on AoE, all you need is to buy some EtherDrivers and connect them to the Ethernet, where a host server(s) is also connected. The host server, running Linux, can then load AoE driver and can access those target EtherDrive blades by /dev/etherd/. Each EtherDrive blade follows the naming scheme of shelf and slot number represented in Linux as /dev/etherd/eX.Y, where X and Y represent shelf and slot number respectively. So, for example, blade in shelf 0 and slot 4 can be accessed from the host system as /dev/etherd/e0.4. In this way, you can access target AoE disks as a standard local block device in Linux and the driver handles all the network communication. You can then create filesystem on EtherDrive blades, which then can be mounted in a normal way and accessed. Moreover, you can also create partitions, configure RAID, create network attached systems and backup storage servers on these target disks.

In case you are not interested in buying EtherDrive blades from Coraid, you can export hard disk(s) of a Linux machine(s), which can then be accessed over AoE by other machines. This can be done by a program vblade, which runs as AoE server and exports storage disks. Other machines that want to access these target storage require AoE kernel driver and can access by /dev/etherd/. There are two separate implementations available of vblade, in user space (http://sourceforge.net/projects/aoetools/) and kernel space (http://lpk.com.price.ru/~lelik/AoE/).

Conclusion

AoE is a simple yet very flexible protocol that can be used to build SANs at a very low cost. It does not incoporate IP, UDP and TCP and thus is not routable, which provides low complexity and low overhead. If you want to build a storage network for local access only, then AoE is definitely useful for you.

References

http://www.coraid.com/documents/EDProductDescription.pdf

http://www.coraid.com/documents/AoEDescription.pdf

http://www.coraid.com/documents/AoEr8.txt

http://www.linuxjournal.com/article/8149

About the Author

Muhammad Hammad is an experienced IT professional from Pakistan. He is currently the Chief Technology Officer for an Open Source Networking Stealth Startup. Prior to co-founding the startup, Hammad held the position of GM of Enterprise Data Networking at Spliced Networks.

Secure Users, Not Ports

Imagine having your corporate resources move with you, securely, everywhere you go. In public places like airports, hotels, coffee shops and cafes, at home, or in a partner's office.

With Aruba Wireless Networks' Mobile Edge Architecture, it is possible. Our identity-based technology secures users and ports. Your wireless connection is more secure than your wired connection, and you are more secure wherever you are, even outside your office.

Find out more about Aruba's Mobile Edge solutions at **www.arubanetworks.com/solutions**

Aruba's Mobile Edge moves.

ARUBA™

The **Mobile Edge** Company

layer 2 security testing with yersinia

LAYER 2 SECURITY IS OFTEN OVERLOOKED BY VENDORS AND ADMINISTRATORS
YERSINIA IS AN OPEN SOURCE TOOL DESGINED TO HELP TEST LAYER 2 PROTOCOLS ON YOUR NETWORK

by John Buswell

Open source network security tools are usually interesting, one in particular that this article focuses on is called Yersinia. Yersinia is unique in that it focuses primarily on Layer 2, an area often forgotten by both administrators and vendors alike.

So what is Layer 2?

Essentially, when we're talking Layer 2, we're talking Ethernet. Layer 2 refers to the Data Link Layer in the OSI model. If you're dealing with MAC addresses, and not IP addresses, then you're in Layer 2 land. Yersinia focuses security testing on this critical layer by looking at a number of different protocols. Yersinia looks at Spanning Tree Protocol, Cisco Discovery Protocol, DHCP, Hot Standby Router Protocol (HSRP), Dynamic Trunking Protocol, 802.1Q, 802.1X and VLAN Trunking Protocol (VTP).

So what are we testing?

The protocols mentioned above might not be ones you are familiar with. While most are supported under Linux, most of them are typically found on routers and switches in more complex network configurations. If you're using Yersinia, you're likely either a network administrator conscience of the security implications at Layer 2, or an engineer at a vendor looking to improve Layer 2 security in a switch or router type device.

Switches and Routers

For the purpose of this article, we'll be testing a Netgear Layer 3 switch and a Cisco Route Switch Module in a Cisco Catalyst 5505. Spanning Tree, 802.1Q, DHCP and 802.1X we are testing on the Netgear, and HSRP, CDP and VTP we are testing on the Cisco. We are testing DTP on both devices.

Be Smart

It is not a good idea to run Yersinia against your live network, unless you know exactly what your doing and you are trying to replicate a specific problem. In which case, Yersinia might be useful in triggering a switch to behave in a certain manner. Ideally Yersinia has its greatest use in QA labs, support labs and engineering labs, where you can isolate attacks to test networks.

Getting Yersinia

Our Linux test system is an AMD64 X2 based Fedora Core 5 server. Yersinia is available from http://www.yersinia.net. Our system was missing libnet and libnet-devel, a quick round of installs with yum and we are ready to build. We used the nightly snapshot of Yersinia so that we had the latest release.

Building Yersinia

The usual autoconf method:

```
[root@] # tar zxvf yersinia-snapshot.tgz
[root@] # cd yersinia
[root@]# ./configure –with-ncurses
[root@]# make && make install
```

Running Yersinia

Yersinia can run in command line mode as well as a GTK graphical version. The snapshot we tried, the GUI was a little unstable, but it worked pretty well. The GUI uses an edit mode, which allows you to modify the attributes that are used. From the GUI, there are tabs for CDP, DHCP, 802.1Q, 802.1X, DTP, HSRP, ISL (Inter Switch Link), STP, VTP and an application log. Since the GUI was a little unstable, could be our 64-bit Fedora Core 5 as well, so we'll stick with the command line. Yersinia also has a pretty slick ncurses interface that you can access with yersinia -I.

Yersinia Ncurses

The console based ncurses mode, gives you interactive control over Yersinia. It is pretty straight forward, press h for the help screen and you can learn the application within a few minutes. The 'e' key allows you to edit the packet fields, 'x' starts the attack, 'l' gives a list of active attacks, 'K' kills off active attacks, and 'g' allows you to switch between protocol screens. This interactive mode is the fastest way to get up and running with Yersinia. Once you've figured out the options, you can easily put together the command line equivalent. You can capture data to a file, which will then enable you to use something like WireShark (formally Ethereal) to perform analysis on your tests.

Spanning Tree Protocol (STP)

Spanning Tree is a link management protocol that provides multiple path redundancy. When a switch has multiple paths, the preferred link (based on cost) is placed in FORWARDING while the redundant link is placed in a BLOCKED mode. Spanning Tree has been discussed in depth previous in o3. STP uses Bridge Protocol Data Units (BPDU) to exchange information between bridges. STP supports three types of BPDUs – Configuration Change, Topology Change Notification and Topology Change Acknowledgment. There are variations of Spanning Tree – STP, RSTP (Rapid Spanning Tree) and MSTP (Multiple Spanning Tree). Yersinia supports all three types and can send both Configuration and TCN BPDUs.

Yersinia suggest 7 different STP related attacks, these are:

- Sending RAW Configuration BPDU
- Sending RAW TCN BPDU
- DoS sending RAW Configuration BPDU
- DoS sending RAW TCN BPDU
- Claiming Root Role
- Claiming Other Role
- Claiming Root Role dual home (MITM)

Launching these attacks is pretty easy with Yersinia. Simply press 'g' and select STP. Select 'e' and fill in some values for your attack, perhaps spoof the MAC of a valid switch on your network. Then press 'x' to select the attack you wish to start. The attacks that perform Denial of Service (READ: THIS COULD BRING DOWN YOUR NETWORK) attacks are marked with an X under the DoS heading. Yersinia allows you to edit the following packet fields:

- Source and Destination MAC addresses
- STP ID
- STP Version (0 = STP, 1 = RSTP, 2 = MSTP)
- Type (Conf, Conf (MSTP / RSTP), TCN)
- FLAGS (TC, TC ACK, Proposal, Learning, Forwarding, Agreement)
- RootID
- Pathcost
- Bridge ID
- Port
- Age
- Max
- Hello
- FWD

As you can see it offers a comprehensive and very simple way to spoof an STP BPDU. The DoS features are extremely good, a quick conf DoS attack generated a 62MB packet capture file in under 5 seconds.

This type of tool is excellent for testing layer 2 switch products, it can be used to send normal STP BPDUs flagged as RSTP or MSTP, or setting flags that regular STP switches wouldn't understand to see if they crash or simply ignore the packets. Running these kinds of tests on equipment you are evaluating or preparing to place into production can locate problems BEFORE someone else tries it on your production network.

Cisco Discovery Protocol

CDP is a proprietary Cisco protocol that is used to identify and locate other Cisco devices on a network. Yersinia supports three CDP attacks:

- Sending a RAW CDP packet
- DoS flooding CDP neighbors table (this is fun)
- Setting up a virtual device

Yersinia allows you to modify:

- Source and Destination MAC addresses
- Version
- TTL
- Checksum

There isn't a lot you can do with CDP, you can see what happens if you try to fill the CDP table on a neighboring router, and you can see how Cisco devices react to spoofed CDP packets, ones with bad checksums or weird TTLs. In our case, we did manage to DoS attack the RSM module fairly quickly.

Dynamic Host Configuration Protocol

DHCP is used to assign IP addresses and other configuration information to clients on a network. DHCP is used by practically every ISP out there. So let me give you a quick warning, DO NOT, regardless of how “cool” you think it might be to test the security of your cable or DSL providers DHCP servers, they don't want you to try, its not cool, and you're likely to lose your access. So just don't go there.

On the other hand, while I worked at Nortel on the Nortel Application Switch line of products, I used Yersinia to test a DHCP health check feature that I implemented, by using its various attacks and incorrect values to test both the switch and the servers behavior during such an attack. Yersinia not only allowed me to perform testing that would have taken much longer to

script with something like ScaPY or write from scratch in C or Ruby, but to run DoS attacks and send bad packets, and determine what happened during specific network situations. The end result is that the customer base ended up with a far superior feature thanks to Yersinia.

Yersinia supports 4 DHCP based attacks:

- sending RAW DHCP packets
- DoS sending DISCOVER packets (using up the ip pool)
- Setting up rogue DHCP server
- DoS sending RELEASE packets (releasing assigned ips)

As you can see already, plenty of fun to be had here with DHCP.

Yersinia allows you to edit the following fields:

- Source and Destination MAC addresses
- Source and Destination IP addresses
- Source and Destination Ports
- Op codes
- Htype Type (this is usually Ethernet)
- Hlength
- Hops
- Xid (transaction ID)
- Secs (seconds)
- Flags
- CI, YI, SI and GI (IP addresses see RFC 2131)
- Client MAC Address

Yersinia doesn't allow you edit some of the additional DHCP fields, such as messing the magic number, or adding options to change the packet into a DHCP INFORM packet etc.

802.1Q

The IEEE 802.1Q standard provides a means of for multiple broadcast domains to share the same physical network link without leakage between the two. 802.1Q is often an important strategy in providing some security in layer 2 networks. VLANs or Virtual LANs are defined by 802.1Q. VLAN tagging allows multiple VLANs to traverse the same physical link. An office network might have several VLANs, for example the finance department might be on VLAN 666 and the support engineers might be on VLAN 102. Each VLAN has its own IP subnet. Without VLANs and 802.1Q, these subnets would be on the same broadcast domain (layer 2), essentially a HUB. It wouldn't be hard for someone in support to sniff the finance traffic and find out how much other people are being paid. With 802.1Q, both the finance and support VLANs might traverse the same physical links and exist on the same switch, but 802.1Q prevents leakage between the two. Yersinia enables you to test just how secure the 802.1Q implementation on your network really is. In 802.1Q mode, we can edit:

- Source and Destination MAC addresses
- Two sets of:
    - VLAN ID
    - Priority
    - CFI (Canonical Format Indicator, whether the MAC in the frame is in canonical format or not)
    - L2 Protocol (IP, .1Q, ARP, RARP, CDP, VTP, DTP, PVST, LOOP)
- Layer 3:
    - Source and Destination IP addresses
    - IP Protocol (icmp, tcp, udp or ospf)
    - Payload

Yersinia supports three possible attacks, although like with each protocol option in Yersinia, the first one enables you to craft practically any type of malformed or corrupt packet. Leaving the scope of possible testing as wide as your imagination. The attacks supported are:

- Sending 802.1Q RAW packets
- Sending double encapsulated 802.1Q packets
- Sending 802.1Q ARP Poisoning (DoS)

For our 802.1Q test, we wanted to see if the Netgear switch would accept 802.1Q packets for the wrong VLAN on a test port. Next we tested to see how well it would stand up against an ARP poisoning attack (not very well).

802.1X

The 802.1X is a port based authentication protocol, its also used in many wireless networks. Linux supports 802.1X with a Linux supplicant (http://open1x.sourceforge.net/). Yersinia provides two 802.1X attacks:

- sending 802.1x RAW packets
- MITM 802.1X with 2 interfaces

The Yersinia implementation allows you to edit a few EAP related options:

- Source and Destination MAC addresses
- Version
- Type
- EAPCode (REQUEST, RESPONSE, SUCCESS, FAILURE)
- EAPID
- EAPType (Identity, Notification, TLS, MD5, OTP, Token Card, LEAP Cisco)
- EAPInfo field

We sent a barrage of fake 802.1X packets to the Netgear switch, but our 802.1X configuration held up pretty well. The Netgear ignored the bad 802.1X packets, and we were able to get it to exchange some information when we sent the correct data.

Dynamic Trunking Protocol

Yersinia supports two attacks, which can be used to test the trunking / port channel capabilities of a switch:

- Sending RAW DTP packets
- Enabling Trunking

Both attacks are good for checking configurations where DTP might be enabled on all ports by default and not configured properly, allowing anyone to try to create a trunk. Yersinia allows you to edit:

- Source and Destination MAC Addresses
- Version
- Neighbor iD
- Status (ACCESS and TRUNK modes – DESIRABLE/ON/OFF/AUTO and UNKNOWN)
- Type (Native, 802.1Q and ISL modes – 802.1Q/ISL/NATIVE/NEGOTIATED)
- Domain

HSRP

HSRP is Hot Standby Router Protocol, it is a proprietary Cisco protocol, most vendors support a similar open protocol called VRRP (Virtual Router Redundancy Protocol). HSRP works by having a pair of routers in a hot standby configuration, if the active router goes down, then the backup takes over, the end result the IP address configured under HSRP appears to always be up, regardless of a router failure. Yersinia offers some potential for fun at Cisco's expense:

- sending raw HSRP packets
- becoming the active HSRP router
- becoming the active HSRP router with multiple interfaces

Yersinia allows us to edit a complete set of fields:

- Source and Destination MAC addresses
- Source and Destination IP addresses
- Source and Destination Ports
- Version
- Opcode (Hello, Coup, Resign)
- State (Init, Learn, Listen, Speak, Standby, Active)
- Hello
- Hold
- Priority
- Group
- Reserved
- Auth

VIP (Virtual IP, this is the IP thats being made highly available by HSRP)

If you haven't already guessed, the idea here is to check the security of your router. What happens if you tell both routers configured as HSRP that they are backup, can you break the HSRP state machine on the router, can you down the network by trying to force the routers into some unsupported state, do your access lists protecting your HSRP configuration work? Here you can test both the HSRP implementation for things that their QA team might not have thought of, as well as the security of your configuration.

ISL mode

At the time of writing there were no possible ISL attacks, the feature appears to be partially implemented, something we can look forward to in the near future, or if you feel like modifying the code and contributing, its a good place to start.

VLAN Trunking Protocol (VTP)

Yersinia has a nice set of attacks for VTP. VTP is a Cisco protocol that manages additions, deletions and renaming of VLANs on a network-wide basis. Designed to reduce administration, the very description sounds like a security nightmare. With it you can launch:

- Sending RAW VTP packets
- Deleting All VTP VLANs
- Deleting specific VLANs
- Adding a VLAN
- Cisco Catalyst Zero Day attack

This is an excellent tool for testing the security of your VTP configuration. Basic tests should include making sure devices on your network that aren't supposed to be able to add and delete VLANs can't, checking to see

what happens with bad VTP packets (will it crash your current implementation).

Yersinia allows you to edit the following VTP related fields in the packets it sends out:

- Source and Destination MAC addresses
- Version
- Code (Summary, Subset, Request, Join)
- Domain
- MD5
- Updater
- Revision
- Timestamp
- Start value
- Followers
- Sequence

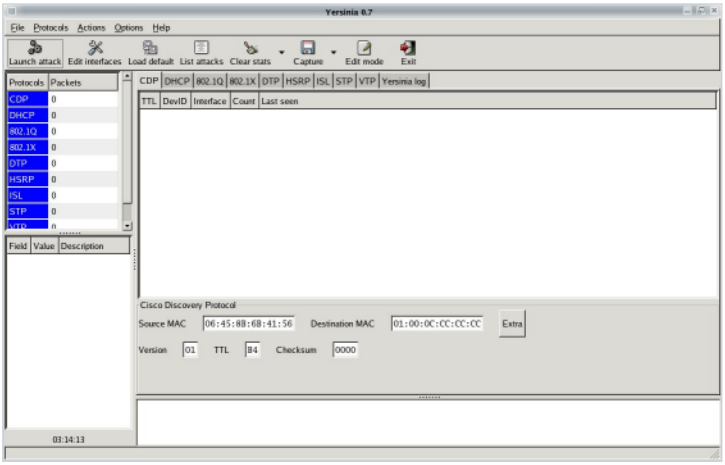


FIGURE 42.1 - YERSINIA GTK INTERFACE

Analysis

Yersinia is capable of splitting capture files into per protocol or lumped together. Writing captures out is critical for analysis during or after the tests. A GUI packet analysis program such as WireShark (www.wireshark.org) is a good choice for inspecting the results. Yersinia is capable of running multiple attacks at the same time, and its an excellent testing tool. The type of analysis you do really depends on what your goal is for testing. If you are looking for a specific response, you should be looking for the MAC or IP address of the responding device, then inspect the response. In most cases, you will want to see NO response to malformed or just plain wrong packets.

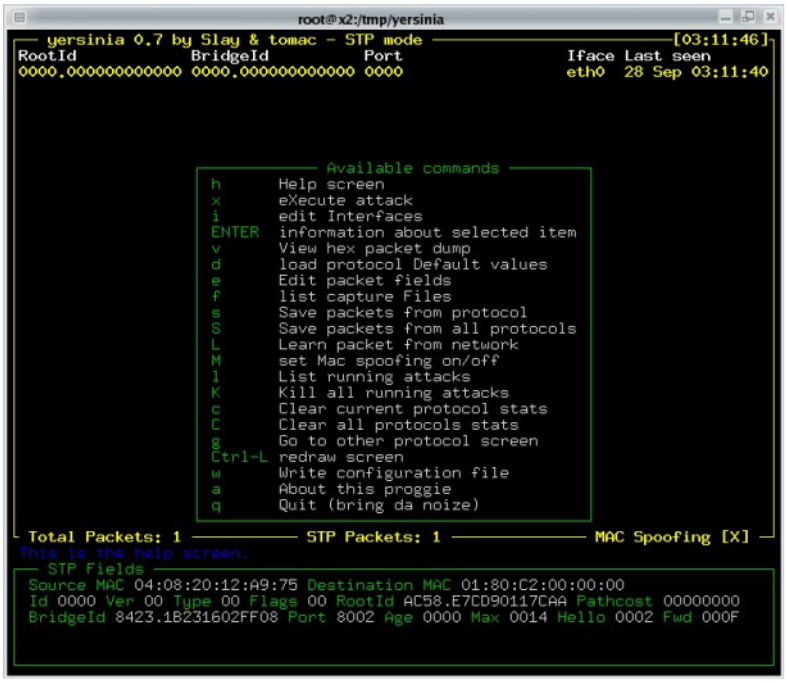


FIGURE 42.2 - YERSINIA CURSES INTERFACE

Conclusion

Yersinia is an awesome tool, although it does have some potential for abuse. If for no other reason, any system or network administrator out there should look at this tool, and be familiar with the Layer 2 attacks. Taking steps to make sure specific attacks are not possible on segments of your network is probably a good idea, and since layer 2 security is often overlooked by administrators and engineers, as Yersinia continues to add features, we can expect to find more and more vulnerabilities. Yersinia is out there, ignoring it won't help you, using it to protect your network before someone uses it to attack your network is a good idea.

>THIS IS **THE WAY**

TO ENSURE 99.999% RELIABILITY.

Hundreds of millions of wireless calls, billions of stock exchange transactions, 80% of the top 100 banks in the U.S., even the U.S. Department of Defense depend on Nortel.™

You'll find us wherever secure, reliable data and voice communications are critical.

>THIS IS NORTEL™

www.nortel.com/enhance

Reported customer availability metrics across Nortel Networks popular products exceeds 99.999%, August 2004.

deploying t1s with linux

WAN LINKS PLAY AN IMPORTANT ROLE FOR BUSINESSES, CISCO IS THE USUAL CHOICE FOR T1 WAN LINKS BUT DEPLOYMENT UNDER LINUX IS POSSIBLE, CHEAPER AND OFFERS BETTER SECURITY AND FLEXIBILITY

This month we look at migrating access (edge) routers from Cisco to Linux. A T1 is the standard connection type for reliable, fast and cost effective Internet access for businesses in America. In Europe the connection is called an E1, and in Japan J1. The bandwidth varies between the different types due to the channels available on each link. For the purpose of this article, we will focus on T1, however the equipment we describe in this article can also run on E1 links.

A T1 provides 1.54Mbps of bandwidth, typically a T1 will cost more than business class DSL or cable connects. However, a T1 is typically far more reliable, and usually comes with a much better Service Level Agreement (SLA). An SLA typically describes the minimum performance the customer can expect from the network. If performance drops below that level, the customer has recourse with the provider. Likewise, an SLA usually has a service response time, and the connection is monitored. If the link goes down, the ISP is contracted to respond and repair within a specific amount of time. The SLA usually specifies whether the customer is entitled for a refund if the connection is down for any lengthy amount of time. A T1 is a good option for mission critical applications, because it is far more reliable than a regular DSL or Cable connection.

Unlike DSL connections, the provider doesn't typically supply the router, instead the customer most purchase, install and configure the router. Most providers will sell you services where they sell you, install, configure and manage the router. You will typically pay a premium for this type of service. If the provider is supplying a Cisco router, and they are providing firmware upgrades, it may be worth the initial installation fee over time, if they are going to upgrade the router for you. With Cisco you need to have a maintenance contract called a Smartnet Contract in order to obtain the firmware for your Cisco router, no exceptions. The cost over a few years, may make any installation fees seem minimal, depending on the type of Cisco router you have. In fact, it may well be because of the cost of upgrading an existing Cisco router that you are reading this article.

In our case, we had a Cisco Route Switch Module (RSM), which is basically a Cisco 7500 router on a module “stick” for the Cisco Catalyst 5500 switching platform, with a VIP2-40 module.The VIP2-40 allows you to plug Cisco Port Adapter Modules into the Cisco Catalyst 5500 for use with the RSM. In our case, we had a PA-4T, quad port T1 card that was connected to an Adtran CSU/DSU. This platform worked fine, but the Catalyst 5500 is a huge piece of equipment that isn't exactly friendly on the electricity bill, after we upgraded from Fast Ethernet to Gigabit Ethernet network-wide, we needed to phase out the 5500.

Selecting a Card

There are a number of T1 cards available that work under Linux. We chose the Sangoma Technologies A101 card which is a single port T1 card that supports HDLC, and has an integrated CSU/DSU. This meant we had at least one piece of equipment less to be concerned about, as we no longer needed the Adtran. Sangoma sell dual and quad port T1 cards. The A101 is reasonably priced, we purchased ours from iFax (www.ifax.com) for $474.00, plus shipping. The A101 is a 32-bit PCI card.

Installation

Installation is simple, we installed the A101 into a Dell 2450 running Gentoo Linux 2006.0. Our installation of Gentoo was running a 2.6.16 kernel, we downloaded the latest Sangoma Wanpipe stable drivers from Sangoma's website. Installation was trivial, simply unpacked the driver tarball, ran ./Setup install, followed the instructions which amounted to hitting enter and answering yes to a few questions.

The driver installation was extremely smooth. Configuration was just as easy with Sangoma's wancfg utility. Simply run wancfg, and you are presented with a nice curses based configuration tool.

First select

> Create a new configuration File
> wanpipe2.conf – does not exist

We chose to select from detected cards, and it found our AFT-A101u card without any problems. They Physical Medium was set by default to T1, if you're in Europe, you'll want to change this to E1. Our T1 uses B8ZS decoding with ESF framing, this is pretty standard, and all the defaults worked reliably for us.

After Hardware Setup, you need to define a protocol.

If you're upgrading from a Cisco router, then it is very likely your current setup is using Cisco HDLC (CHDLC) as the protocol. You can check this by running the show interface command on your Cisco router, you want to look for the Encapsulation line.

Serial1/3 is up, line protocol is up
Hardware is cyBus Serial
Description: BQWT1-xxxxx-9xxxxxx1
Internet address is 10.14.21.166/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 10/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Restart-Delay is 0 secs
Last input 00:00:04, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 63
Queueing strategy: weighted fair
Output queue: 0/1000/64/63 (size/max total/threshold/drops)
Conversations 0/25/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 1158 kilobits/sec
5 minute input rate 66000 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
4599972 packets input, 3258648379 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
2 input errors, 2 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4680418 packets output, 1326559419 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 117120 output buffers swapped out
1 carrier transitions
RTS up, CTS down, DTR up, DCD up, DSR up

Cisco HDLC requires very little configuration, the defaults worked fine for us. If you use another type of encapsulation, you can typically pull the information you need from the Cisco router to configure the Sangoma card. You should now return to the previous menu where you'll have:

Hardware Setup --> A101/2
Protocol ----------> CHDLC
Interface Setup ----> 1 defined

Select the Interface Setup, you'll see your interface w1g1chdl, select it. We left the mode as WANPIPE, this is fine if you're just swapping out a regular Cisco router and don't need to do anything fancy. Configure the IP information, your local IP, you can pull from the Cisco router using ***show ip interface brief*** command. Simply look for the interface that corresponds to your T1 on the Cisco router, and on the same line you'll find your IP address. The Point-to-Point address is the IP address on the other side of the T1, your providers router. Typically an ISP will use a /30 (2 host) subnet, so you can easily determine the remote IP from looking at the assigned ip on the cisco to that interface. Alternatively, you can look at the routing table on the cisco (show ip route) which should list the remote end as the gateway of last resort, but that might not always be the case. If you are uncertain, check with your provider. Enabling Dynamic Interface Config, will mark the interface up and down as the link goes up and down, this makes things a lot easier to debug, so enable it.

Finally, head down to the Advanced WANPIPE options menu, and wrap up the configuration. Save as you exit out, and your configuration is complete. Now, to test it, simply plug the Sangoma card into your smart jack with the supplied T1 cable, and run wanrouter start wanpipe1. Where wanpipe1 was the configuration we selected to use. If you used a different configuration slot, adjust the command accordingly.

Now, in dmesg you should see something along the lines of :

wanpipe: WANPIPE Modules Unloaded.
WANPIPE(tm) Hardware Support Module Stable 2.3.3-3 (c) 1994-2005 Sangoma Technologies Inc
WANPIPE(tm) Interface Support Module Stable 2.3.3-3 (c) 1994-2005 Sangoma Technologies Inc
WANPIPE(tm) PPP/Cisco HDLC Protocol Stable 2.3.3-3 (c) 1994-2005 Sangoma Technologies Inc
WANPIPE(tm) Multi-Protocol WAN Driver Module Stable 2.3.3-3 (c) 1994-2005 Sangoma Technologies Inc
wanpipe: Probing for WANPIPE hardware.
ACPI: PCI Interrupt 0000:00:08.0[A] -> GSI 22 (level, low) -> IRQ 21
wanpipe: AFT-A101u T1/E1 card found (HDLC rev.25), cpu(s) 1, bus #0, slot #8, irq #21
wanpipe: Allocating maximum 1 devices: wanpipe1 - wanpipe1.
WANPIPE(tm) Socket API Module Stable 2.3.3-3 (c) 1994-2005 Sangoma Technologies Inc
NET: Registered protocol family 25
af_wanpipe: Registering Wanpipe API Socket Module

WANPIPE(tm) L.I.P Network Layer Stable 2.3.3-3 (c) 1995-2004 Sangoma Technologies Inc.
WanpipeLIP: Protocols: FR PPP CHDLC LIP_ATM
wanpipe1: Starting WAN Setup

Processing WAN device wanpipe1...
wanpipe1: Locating: A101/2 card, CPU A, PciSlot=8, PciBus=0
wanpipe1: Found: A101/2 card, CPU A, PciSlot=8, PciBus=0
ACPI: PCI Interrupt 0000:00:08.0[A] -> GSI 22 (level, low) -> IRQ 21
wanpipe1: AFT PCI memory at 0xFE042000
wanpipe1: IRQ 21 allocated to the AFT PCI card
wanpipe1: Initializing for SMP
wanpipe1: Starting AFT Hardware Init.
wanpipe1: Enabling front end link monitor
wanpipe1: Hardware Adapter Type 0x41 Scurity 0x00
wanpipe1: Security 1 Line UnCh
wanpipe1: Configuring PMC COMET T1 Front End (Port 1)!
wanpipe1: All channels enabled
wanpipe1: Configuring Device :wanpipe1 FrmVr=25
wanpipe1: Global MTU = 1500
wanpipe1: Global MRU = 1500
wanpipe1: RBS Signal = Off
wanpipe1: FE Ref Clock = Osc
wanpipe1: TDMV Span = Not Compiled
wanpipe1: Configuring Interface: w1g1
wanpipe1: UsedBy :STACK
wanpipe1: MRU :1500
wanpipe1: MTU :1500
wanpipe1: HDLC Eng :On
wanpipe1: Timeslot Map :0xFFFFFFFF
wanpipe1: DMA MRU :2048
wanpipe1: RX DMA Per Ch :10
wanpipe1: Net Gateway :No
wanpipe1: Registering LIP w1g1chdl -> w1g1
w1g1chdl: Running in WANPIPE mode
w1g1chdl: Sync CISCO Configuration
w1g1chdl: Keep Alive Timer :5
w1g1chdl: Keep Alive Cnt :10
wanpipe1: T1 disconnected!
wanpipe1: T1 connected!
w1g1: Lip Link Carrier Connected!
w1g1chdl: protocol up
w1g1chdl: Lip Dev Prot State Connected!

Finally, **ip addr** should display the information for the T1. you'll see two interfaces, one for the T1 and one for the Cisco HDLC encapsulation over the T1:

7: w1g1: <NO-CARRIER,POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 100 link/ppp

8: w1g1chdl: <NO-CARRIER,POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 100 link/ppp
inet 10.14.21.166 peer 10.14.21.165/30 scope global w1g1chdl

Here you see the links are down, the w1g1 comes up first, then after about 30 seconds, the w1g1chdl will come up, the NO-CARRIER is removed. As you see here :

19: w1g1: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 100
link/ppp
20: w1g1chdl: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 100
link/ppp
inet 10.14.21.166 peer 10.14.21.165/30 scope global w1g1chdl

Conclusion

Overall, the total time to switch over from the Cisco router to the Linux box with the Sangoma card was less than 30 minutes. This included the hardware installation of the Sangoma card into the Linux router. A few iptables commands to configure NAT, or IP masquerading depending on your IP allocation, and perhaps a few iptables commands to duplicate functionality of the Cisco Access Control Lists on your cisco router, and you're done.

About the Author

John Buswell is Editor in Chief of this magazine, and Chief Technology Officer of Spliced Networks LLC. You have the opportunity to meet John Buswell at Ohio LinuxFest on Saturday September 30th 2006. For more information visit http://www.ohiolinux.org.

Is your Linux Appliance Project headed down...

... the wrong Path ?

Get an AppOS developer account today at

http://www.splicednetworks.com

Spliced Networks

4820 Fisher Road
Athens, OH 45701

1.408.416.3832

http://www.splicednetworks.com

deploying open source databases: mysql

OUR FIRST SERIES OF "FOCUS ON" ARTICLES LOOKS AT OPEN SOURCE DATABASES
MYSQL IS THE MOST POPULAR OPEN SOURCE DATABASE SO ITS ONLY FITTING THAT WE LOOK AT IT FIRST

by John Buswell

This new focus on column has been in the pipeline for some time now, and there is no better place to start than with MySQL. MySQL is the most widely used open source database solution available today. MySQL is developed by a company called MySQL AB, which offers commercial products based off MySQL and a variety of support services. Many web based open source solutions utilized MySQL, in fact the “M” in the infamous LAMP term stands for MySQL. LAMP describes the Linux, Apache, MySQL, PHP suite of open source applications that are pieced together to provide a high performance web application framework. In this article, we will introduce you to MySQL, and walk you through the basic installation and configuration. Next issue, we will do the same with PostgreSQL, another widely popular open source database system. PostgreSQL is often preferred by developers because of the less restrictive licensing placed on PostgreSQL, its completely free.

Introduction

For the purpose of this article, we look at MySQL 5.0.22 [Community Edition]. As with many open source projects backed by companies, MySQL's community edition has a faster life cycle than its commercial counterpart. This is not necessarily a bad thing, if you can perform testing, and work out potential bugs in-house. Having used MySQL for a number of years, we've yet to run into a major problem as a result of using the Community version of the database system. Typically, the community edition hasn't gone through as rigorous a testing cycle as the commercial version, and its likely there is less of a decision making process behind what patches are worked into the community version. If you are running something that is highly business critical, it is well worth looking at the product offerings from MySQL AB, if on the other hand, you have the talent in house, it can offer significant savings to roll with the Community Edition.

MySQL AB offers a SAP certified database called MaxDB. MaxDB contains some features that are not in MySQL, while MaxDB may not support all the platforms that MySQL supports. Key differences in functionality revolve around the network protocol used for client / server communications, and MaxDB is distributed with text, graphical and web based interfaces. So if you don't need the SAP certification, in the majority of cases, MySQL Community Edition will work just fine for your project.

Setting up the environment

MySQL 5.0.22 is available from a number of mirrors located through the Developer Zone link on the MySQL website (http://www.mysql.com). For the purpose of this article, we used an AMD64 2.0GHz system running Gentoo 2006.0 with 1GB ram.

Once downloaded, simply untar the archive with *tar zxvf mysql-5.0.22.tar.gz*.

Before we can build the source, we need to add a user and group for mysql. The default is to use mysql, however its not a bad idea to use something different to make it a little harder for potential malicious users. It is also useful if you want to run different versions of mysql on the same system or different instances of mysql on the same system. We will use the default for this article. Since we're building from source, we also need to put mysql in a place where it won't get confused with files from the linux distribution we are using. You can put mysql anywhere you please, but for this article, we're going to place it in /opt/db/mysql. We're using /opt/db/mysql because next issue, we will be using /opt/db/pgsql to store PostgreSQL.

```
# groupadd mysql
# useradd -g mysql -d /opt/db/mysql mysql -m
```

MySQL can provide client / server transport over SSL encrypted connections. To do this, it needs openssl. Now you can use openssl that came with your Linux distribution, but I find that its often better when building code from source, to use your own set of libraries. So for the purpose of this article, I built openssl 0.9.8b:

```
# wget http://www.openssl.org/source/openssl-0.9.8b.tar.gz
# tar zxvf openssl-0.9.8b.tar.gz
# cd openssl-0.9.8b
# ./config –prefix=/opt/db/openssl \
 –openssldir=/opt/db/ssl zlib-dynamic \ shared
```

```
# make
# make test
# make install
```

As you can see above, I've installed openssl into /opt/db/openssl. Next issue, we'll build postgresql against the same version of openssl. The ssl directory is where the ssl cert and other configuration data is stored, this shouldn't be confused with the prefix directory which is where the binaries, libraries and includes for openssl are installed. These should be different paths!

Building from source

Now that the environment is ready to go, it is time to compile MySQL. While MySQL AB have big warning signs about compiling from source, following the instructions below will give you a production safe environment.

```
# CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 \
 -felide-constructors \
 -fno-exceptions -fno-rtti" ./configure \
–prefix=/opt/db/mysql/ \
--with-mysqld-user=mysql \
–with-unix-socket-path=/tmp/mysql.sock \
–enable-assembler \
--with-openssl=/opt/db/openssl/ \
--with-openssl-includes=/opt/db/openssl/include \
--with-openssl-libs=/opt/db/openssl/lib \
--enable-thread-safe-client
```

If you don't want to use openssl, you should include the line **–with-mysqld-ldflags=-all-static**, and remove the openssl lines. The **–enable-thread-safe-client** is used so that the extra-tools for mysql are built. If you don't need them, you can drop that line. Scroll up through the output from the configure command, make sure it didn't produce any errors if your environment is missing something or too old for the current version. If it does produce errors about packages being too old, simply follow the instructions for your distribution to update it. To build mysql, we simply run **make**, and grab a cup of coffee.

```
# make
# make install
# mkdir -p /opt/db/cfg
# cp support=files/my-medium.cnf /opt/db/cfg/my.cnf
# ln -sf /opt/db/cfg/my.cnf /etc/my.cnf
```

The last three lines above copy the default medium server config to /opt/db/cfg, and symlink /etc/my.cnf to this location. The reason we are doing this is to centralize our db configurations for easier management, but also allows us to grant a db administrator write permissions to /opt/db/cfg without having to give them access to /etc.

Since we built custom libraries, we need to update /etc/ld.so.conf with the paths for ssl (/opt/db/openssl/lib) and mysql (/opt/db/mysql/lib/mysql/). Simply add these to /etc/ld.so.conf and run ldconfig. If you are running Gentoo though, you'll need to add these to a file in /etc/env.d/ and run env-update. Otherwise you'll lose the path next time Gentoo runs env-update.

Now, we need to do some configuration and setup. Since we're just introducing MySQL we're not going to look at doing anything complex such as placing MySQL in a chroot environment. However, after the PostgreSQL introduction, we will be looking at how to chroot both database systems.

```
# cd ~mysql
# bin/mysql_install_db –user=mysql
# chown -R root .
# chown -R mysql var
# chgrp -R mysql .
# bin/mysqld_safe –user=mysql &
```

The above commands set some default permissions, populate basic database information and the last line starts the mysql server. By default, mysql has no administration password, so we need to set one up. One of the bad things that mysql recommends that you do is issue the password command to mysqladmin from the command line. If you're using a shell that contains command history, someone can just run history later on and find out what you set the password to. While you might argue that someone would need root, but its just a bad habit to get into leaving passwords in the command history. To get around this problem, we simply put together a script, which we can delete afterwards.

```
# nano -w ./dbpass.sh

#!/bin/bash
/opt/db/mysql/bin/mysqladmin -u root password \
  'mynewdbpass'
/opt/db/mysql/bin/mysqladmin -u root -h \
  myhostname.domain password 'mynewdbpass'
rm -rf ./dbpass.sh
```

```
# chmod 700 ./dbpass.sh
# ./dbpass.sh
# ls -la | grep dbpass
```

The last line should return nothing, and your database administration passwords have been changed without record on the local file system.

Since we'll want to optimize mysql later, its always a good idea to get a baseline on the server performance, so we're going to run the sql-bench benchmarking tools that come with mysql. You'll need perl and the DBI module for perl, if you're running Gentoo, you can simply do emerge DBI. You will still need the perl DBI/mysql.pm module, the easiest way (at least in Gentoo) without having Gentoo pull down mysql again, is to use the CPAN shell. If its the first time you've run this, it'll ask some questions, just customize it as you see fit. We stored our cache in /var/spool/perl/CPAN. If you did a fresh install of Gentoo, you'll probably want to emerge unzip, ftp, lynx, and gnupg. Don't forget to prefix with USE="-X" to avoid installing all the GUI stuff you probably don't need.

```
# PATH=$PATH:/opt/db/mysql/bin
# export PATH
# perl -MCPAN -e shell

cpan> get DBD::mysql
cpan> make DBD::mysql
cpan> test DBD::mysql
cpan> exit

# cd /var/spool/perl/CPAN/build
# ls

DBD-mysql-3.0006

# cd DBD-mysql-3.0006
# make install
# cd ~mysql
# mysql –user=root mysql -p
  Password: ********

mysql> GRANT ALL PRIVILEGES ON *.* to
          'test'@'localhost' IDENTIFIED BY 'test' WITH
             GRANT OPTION;

mysql> FLUSH PRIVILEGES;

mysql> quit

# cd sql-bench
# perl run-all-tests –server=mysql –user=test \
    –password=test –log
# mysql –user=root mysql -p
  Password: ********

  mysql> DROP USER test;
  mysql> FLUSH PRIVILEGES;
  mysql> quit
```

You'll probably get some errors about permissions with the DBD test, just ignore those.

Networking

In some situations you'll have the mysql server on the same system as the application needing to use the database. This type of scenario is typical if you have a single colocated server or you are leasing a dedicated server from a provider. In such cases, you don't need the added security risk of running MySQL on a TCP port, instead you can use unix sockets for communication with MySQL. To configure MySQL for unix sockets simply edit my.cnf, the modified lines are in bold in the config segments below:

```
[client]
#port = 3306
socket = /tmp/mysql.sock

[mysqld]
#port = 3306
socket = /tmp/mysql.sock
skip-networking
```

Simply configure your application to use the sockets method, this varies from application to application, and is often detected automatically. There is some magic needed if you're running mysql in a chroot, which we will discuss in a later article.

In most enterprise environments though, you will likely want to run a dedicated database server or cluster of database servers. MySQL 5.x provides a variety of features for running database servers in clusters, several problems with data integrity, especially write integrity across multiple servers is a key problem for clustered database solutions. Our build of mysql didn't compile cluster support, we will discuss database clustering and load balancing in a future article. The goal of this column is to build up to more complex solutions, as you would scale a solution in a real world deployment, thus allowing us to address migration issues along the way.

MySQL uses TCP connections to perform client/server operations between the database server and the application using the database. If you compiled MySQL with SSL support, then it is important to use SSL encrypted sessions for these communications. There is no point investing in highly secure solutions on your web server, if someone can simply run tcpdump and capture data by intercepting the MySQL traffic between your web server and your database. While an outside attacker might have to compromise your network, there is nothing stopping someone on your local network from collecting data they are not supposed to have access to using a basic packet capture tool.

Dedicated Network or VLAN

Placing MySQL traffic on a dedicated physical network or VLAN offers an additional layer of security, while providing an important foundation for scaling the network up at a later stage. If the deployment has only a small number of servers that need to access the database server, and if you anticipate that your SQL traffic will grow rapidly over the next couple of years, it is a worthwhile investment to purchase some additional GbE (Gigabit Ethernet) adapters and deploy a separate physical network for your database traffic.

What do we mean by separate physical network? Simply put, each server that needs access to the database, and each database server would have a dedicated Fast Ethernet or GbE adapter connected to either a dedicated switch or a dedicated VLAN on a switch. All database traffic would exist on a dedicated IP subnet (i.e 10.33.06/24), which would not be routed outside of the VLAN. It is a wise precaution to block the database IP subnet at your firewalls to prevent any possible leakage. Likewise using iptables on each of your servers and database servers to block all non-database and non-database IP subnet traffic on the interface attached to the database VLAN is a good idea.

This type of deployment has advantages as you add features to your network. For example, an IDS sensor can be attached to the database VLAN. Since the VLAN only passes SQL traffic, the IDS sensor can be fine tuned and optimized to look specifically at SQL data. When you add replication to your database deployment, you already have a dedicated high speed network for pushing database traffic between servers. Concerns with network performance, and scalability as your network traffic grows are less of a concern, as the database traffic is not on your production VLANs, and it doesn't have to co-exist with other data on the wire. Optimizations to switch equipment, such as MTU size, can be optimized for database traffic. In the future, should you migrate to a layer 4 switched environment (eg. IP load balancing) for your database services, you simply need to move your database servers behind the load balancers virtual server, and configure the virtual server to match your existing database server's IP. Facilitating fast and simple migration to a load balanced environment.

The cost for a few switch ports, and network adapters is quickly offset by the immediate performance gains, and long term security and scalability advantages. By planning your network for future scalability, whether you're starting with one server and one database server, will save you from future headaches.

Testing MySQL

By now, the sql-bench results should be in. They are stored in output/RUN-mysql-*, where * is specific to your kernel, distribution and platform. The test results will vary depending on your hardware, and what the system is doing at the time. Our results were pretty respectable for the hardware we used – insert (683 wallclock secs), select (263 wallclock secs), create (444 wallclock secs) and connect (63 wallclock secs).

Below are a couple of commands that you can use to inspect the configuration, status and database information in mysql. It is a good idea to familiarize yourself with the output, and make sure options which probably contain default values are configured the way you need them.

```
# mysqladmin -p version
# mysqladmin -p variables
# mysqlshow -p
# mysqlshow -p mysql
```

MySQL Anonymous Accounts

Earlier we secured the default superuser “root” account by setting a password. MySQL also includes two anonymous accounts which by default contain no password. There is no legitimate reason for leaving these accounts on the server, when you add applications, open source, in-house or commercial, they should always have their own database and database account. That way, you can separate permissions, and if one application is compromised, only the data from that application is at risk. The MySQL documentation has details on how to set the password on these anonymous accounts, but we're just going to delete them.

```
# mysql -u root -p
  Password: ******
```

```
mysql> DELETE FROM mysql.user WHERE User = '';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

Securing MySQL

If you're using MySQL on a shared data network, with port 3306 open, it is a good idea to use iptables to restrict access to the MySQL port.

```
# iptables -A INPUT -p tcp -s 10.20.30.40 \
                  –dport 3306 -j ACCEPT

# iptables -A INPUT -p tcp –dport 3306 -j DROP

# iptables -A OUTPUT -p tcp -d 10.20.30.40 \
                –sport 3306 -j ACCEPT

# iptables -A OUTPUT -p tcp –sport 3306 -j DROP
```

On the database server itself, simply run the above iptables commands. The examples above will permit MySQL traffic between the database server and 10.20.30.40 (an application server on our network). If you want to add other servers, you can define either the subnet, or each server's IP individually by duplicating the ACCEPT lines (but changing the 10.20.30.40 IP address) and inserting them before the DROP lines.

IP binding

MySQL supports a configuration option called bind-address. This enables you to bind the MySQL server to a specific IP address. The command only permits you to bind to a single address, so you can bind it to an IP such as 10.20.30.10, but then you cannot bind it to localhost at the same time. If you don't need to access the database server via localhost, simply use bind-address to bind it to your database server IP. However, if you need to bind across multiple IP addresses, then you'll have to remove the bind-address configuration option, and use iptables to restrict access. When you remove bind-address it will bind to 0.0.0.0:3306.

Adding and Removing Databases

Almost all of the manual database operations you will do, will be done through the mysql program.

```
# mysql -u root -p
  Password: ******

  mysql> SHOW DATABASES;
+----------------------------+
| Database                   |
+----------------------------+
| information_schema |
|     mysql                  |
|     test                   |
+----------------------------+
3 rows in set (0.00 sec)
```

The USE command allows you to access a database:

```
  mysql> USE test
  Database changed
```

The USE and quit commands do not require a semi-colon at the end of the command. To create a database, use the CREATE command:

```
  mysql> CREATE DATABASE o3database;
  mysql> USE o3database
  mysql> CREATE TABLE test (name VARCHAR(30),
         result VARCHAR(20));
  mysql> SHOW TABLES;
  mysql> DESCRIBE test;
```

The LOAD command can be used to load data into your new table from a file. You can also use the INSERT command to manually insert information into the database. The SELECT command is used to retrieve data from the table. You can remove the database use the DROP DATABASE command:

```
  mysql> USE mysql
  mysql> DROP DATABASE o3database;
```

These commands are SQL, the MySQL reference manual for 5.0 has a good deal of useful information on how to use these commands, and their syntax. Any good SQL book should also cover commands that work with MySQL. In many cases, you'll be using the database server with a web application, which will typically involve importing pre-defined commands into mysql, after creating a new user and database called webapp1, the following command was used to import the database structure provided by the web application we're installing:

```
# mysql -D webapp1 -u webapp1 \
    < /tmp/webapp1-data.sql
```

Then you can use mysql -u root -p and use the GRANT command as we've used above to set the password for the webapp1 user. Next issue, we introduce PostgreSQL, another open source database solution.

Is your data center cramping your style?

Growth always seems like a good idea. An extra processor here—one more server there. Until, all the sudden your data center feels as crowded as a center seat in coach. Let **the Penguin** upgrade you. Penguin Computing introduces BladeRunner™ 4140 the industry's densest Linux blade server. It comes with the AMD Opteron™ HE processor, which offers simultaneous 32- and 64- bit computing. So now you can pack 48 cores into a miniscule 4U of rack space, and optimize your data center. And put that 8GB of PC3200 RAM per blade to work and run your 64-bit apps in a fraction of the space. So go ahead. Stretch your legs. Tilt your seat back. **Love what you do.**

Visit www.penguincomputing.com

AMD Opteron is a trademark of Advanced Micro Devices, Inc
Other names are for information purposes only and may be trademarks of their respective owners.

"Open Source Zero Day Attack Protection"

presented by

John Buswell

Chief Technology Officer

Spliced Networks LLC

http://www.splicednetworks.com

4.00pm EST - Ballroom 2