REVIEWED: PostgreSQL... the advanced database server

O3:

The Open Source Enterprise Data Networking Magazine

Issue 2 / December 2005

http://www.o3magazine.com

Keeping web projects on track and on budget with Ruby on Rails

SCTP vs TCP ..... Transmission Control Protocols Compaired

modsecurity a next generation Open Source web application Firewall

Asterisk / Rails Integration made simple with RAGI

Intrusion Detection SNORT config, extras and RULES

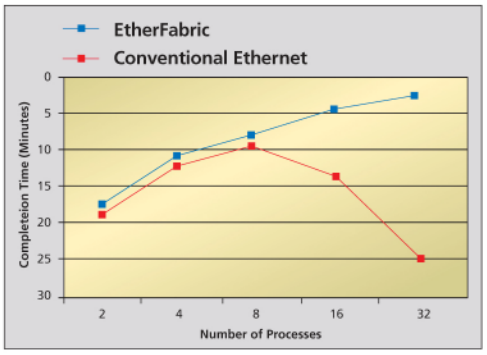
Rapid Web Development for business

ACCELERATE APPLICATION PERFORMANCE!

>> HALF THE LATENCY

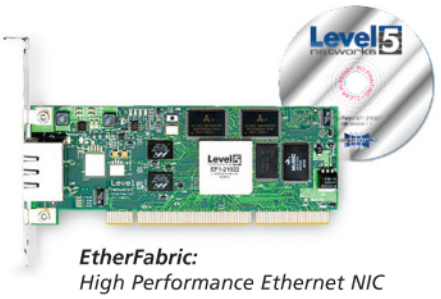
>> TWICE THE BANDWIDTH

>> 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

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

CONTENTS

@O3

INTERNET

Abul Asim M. R. Qarshi looks at AJAX, a solution for refresh-less web applications. AJAX in action on Google Maps.

BUSINESS

James Hollingshead provides a detailed introduction to Rapid Web Development, and the cost savings benefits.

VOIP (Voice over IP)

Asterisk / Ruby on Rails integration with RAGI. An Open Source solution that enables rapid development with Asterisk.

NEXT MONTH

Linux on IBM mainframes
Grids, Clusters and Linux
Porting to the zSeries
Introducing dNMS and more..

SECURITY

A next generation Web Application security solution, providing IDS / Firewall for Web Applications. Supports Apache.

WEB TECH

Keeping web projects on track and within budget using Rapid Web Development tools such as Ruby on Rails and Ruby Gems.

NETWORKING

A look at SCTP, and a comparison with TCP.

The advanced open source database.

Testing Snort, extras and rules..

Reclaim lost time

Levanta Intrepid ™

The world's first Linux management appliance

Plug the Levanta Intrepid ™ into your network and perform the most important Linux management tasks in a fraction of the time you spend now. And gain power and flexibility that you've never had before:

- **Fast & Portable:** Provision servers or workstations practically anywhere, anytime – in minutes. Swap them around, mix it up.
- **Flexible:** Supports commodity hardware, blades, virtual machines, and even mainframes.
- **Out of the Box:** Includes pre-defined templates for servers, workstations, & software stacks. Or create your own.
- **Total Control:** Track any file changes, by any means, at any time. And undo them at will.
- **Disaster Recovery:** Bring dead machines quickly back to life, even if they're unbootable.

Based upon technology that's already been proven in Fortune 500 enterprise data centers. Now available in a box, priced for smaller environments. Just plug it in and go.

30-Day
Money-Back Guarantee

Order online by 2/28/06
Get $500 Off
Enter PROMO CODE: 03M1205

LEVANTA
www.levanta.com
1.877.LEVANTA

WINNER
LinuxWorld
Most Innovative Hardware Solution

Down to business..

WITH OVER HALF A MILLION READERS IN OVER 140 COUNTRIES WORLDWIDE O3 HAS ARRIVED AND NOW ITS DOWN TO BUSINESS...

BY JOHN BUSWELL

Everyone here at O3 would like to thank our readers for taking the time to check out the magazine last month. We would also like to send a special thanks to everyone who sent us suggestions, and a very special thanks to the great Scribus community for pointing us in the right direction with some advanced PDF techniques.

This month you will notice some major enhancements: we have added RSS subscription feeds, a "podcast" style automated RSS 2.0 feed, an announcement mailing list and, most importantly, PDF links. Please keep the suggestions coming, and we will do our best to accommodate as many of the requests as possible.

Last month was a huge success. O3 was read in over 140 countries with more than half a million readers, but most importantly, the community has rallied around O3. The feedback we received shows us that O3 fills a gap which has existed for quite some time in the Open Source world. We realize that the gap O3 fills requires a great deal of responsibility, and we hope you find our commitment to Open Source (and high quality content) is worthy of your time each month.

This month we look at rapid web development and AJAX. While our focus is on Ruby and Ruby on Rails, I want to highlight that Ruby based solutions are not the only open source rapid web development tools available.

James Hollingshead looks at a wide variety of tools including those for PHP, Python and Ruby, as well as a top down view on rapid web development in general. This month's Internet article looks at AJAX and SAJAX and examines Google Maps as an example of what AJAX is capable of. An in-depth look at Ruby on Rails, Integrating Ruby on Rails with Asterisk in our VoIP article, along with a look at SCTP, Postgres and our continued IDS series wrap up this month's articles.

Next month, O3 focuses on "Big Iron" solutions. We take an in-depth look at Linux on the IBM zSeries mainframes, porting applications to the zSeries, VoIP applications you can run on the zSeries and a look at alternatives to mainframes (including grids, clusters and other distributed systems). Issue 3 will continue our IDS series looking at IDS Load Balancing, building secure Linux based appliances and an introduction to a new project called dNMS.

O3 is a media sponsor for Linux Asia 2006, we look forward to the event and sponsoring many other events in the near future.

In conclusion, I would also like to take the opportunity to wish all of our readers a happy and peaceful holiday season from everyone here at O3 Magazine. I found a great charity called ChildsPlay, it is worth a look over at *http://www.childsplaycharity.com*.

O3 Magazine
December 2005
Issue 2

EDITOR IN CHIEF
JOHN BUSWELL
EDITOR@O3MAGAZINE.COM

EXECUTIVE EDITOR
JAMES HOLLINGSHEAD
JAMES@O3MAGAZINE.COM

ARTWORK
JOHN BUSWELL

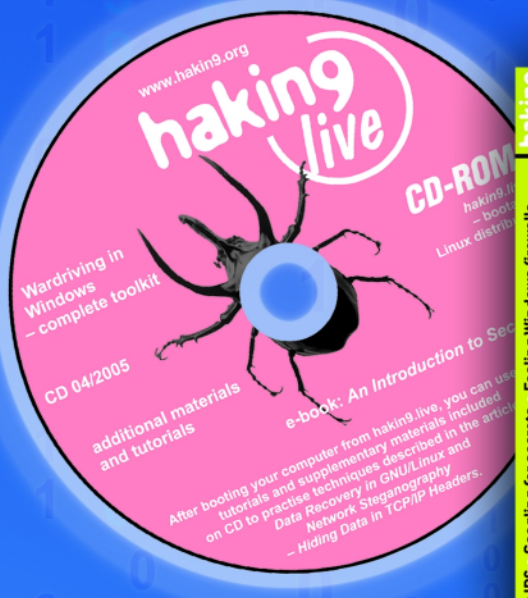
PROOF READERS
GREG JORDAN
SHAWN WILSON
FRANK BOYD
STEW BENEDICT

SALES AND MARKETING
GREG JORDAN
SALES@O3MAGAZINE.COM

SUBSCRIPTIONS
O3 MAGAZINE IS DISTRIBUTED ELECTRONICALLY FREE OF CHARGE BY SPLICED NETWORKS LLC. TO SUBSCRIBE VISIT WWW.O3MAGAZINE.COM.

SOFTWARE
SCRIBUS 1.3.1
GIMP 2.0.5
OPENOFFICE 1.1.2

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

UPCOMING EVENTS

22nd Chaos Communication Congress

december 27 - 30 2005

berlin, germany

http://www.ccc.de/congress/2005

LinuxWorld Expo

february 14 - 17 2006 (mexico city, mexico)

march 28 - 30 2006 (sydney, australia)

april 3 - 6 2006 (boston, united states)

april 20, 2006 (kuala lumpur, malaysia)

april 24 - 26 2006 (toronto, canada)

http://www.linuxworldexpo.com

OOP 06 (Object Orientated Prog. Conference)

january 16 - 17 2006

munich, germany

http://www.sigs-datacom.de/sd/kongresse/oop_2006/index.php

Open Source in the Enterprise

january 23 - 25, 2006

san francisco, california

http://www.marcusevansbb.com/opensource

Linux.conf.au

january 23 - 28, 2006

dunedin, new zealand

http://lca2006.linux.org.au

UPCOMING EVENTS

O'Reilly Emerging Telephony Conference

january 24 - 26, 2006

san francisco, california, usa

http://conferences.oreillynet.com/etel

DITA: Getting Started

january 31 - february 1, 2006

mountain view, california, usa

http://www.comtech-serv.com/workshops/dita.shtml

Linux Solutions

january 31 - february 2, 2006

paris, france

http://www.solutionslinux.fr

have an upcoming event? tell us about it, send email to events@o3magazine.com with details.

FEATURED FUTURE EVENT

LinuxAsia 2006

february 8 - 10 2006

new delhi, india

http://www.linuxasia.net

LinuxAsia is an open source conference and expo which has been held annually in New Delhi since 2004. it is a venue where decision makers, analysts, managers, and technologists from both industry and government come to learn, network, and interact with their open source peers from both India and around the world.

LinuxAsia is being organized by the Electronics For You (EFY) Group (http://www.electronicsforu.com/) and Technetra (http://www.technetra.com/). The conference advisory committee includes members from IIT Bombay, Stanford University, Intel, IBM, Red Hat, Open Source Inititive, Novell, and Intel Capital. O3 Magazine is a media sponsor of LinuxAsia 2006."

DECEMBER OPEN SOURCE REPORT

Welcome to the Open Source Report. This is the section of O3 where we give a brief run-down of the major applications which made releases during the month.

RUBY ON RAILS
http://www.rubyonrails.org/
Release: **1.0**

Rails has now reached a 1.0 release, making it a more solid, stable and polished release over previous versions.

APACHE
http://www.apache.org/
Release: **2.2.0**

The latest release of Apache is the start of a new stable branch. This release has added Smart Filtering, Improved Caching, AJP Proxy, Proxy Load Balancing, Graceful Shutdown support, Large File Support, the Event MPM, and refactored Authentication/Authorization.

SIEGE
http://www.joedog.com
Release: **2.65b1 (beta)**

The latest release of Siege includes several bugfixes and improvements, including improved header handling.

FETCHMAIL
http://fetchmail.berlios.de/
Release: **6.3.0 (stable)**

The latest release of Fetchmail includes a few configuration changes and bugfixes. Some documentation and translation updates have also been made in order to improve stability and protocol conformance, improve bounce and warning messages, and to improve portability.

MILLSTONE
http://www.millstone.org/
Release: **3.1.**0

The latest release of Millstone includes many new features for UI components, better integration with different J2EE environments, and enhancements required for upcoming AJAX support.

XEN
http://www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html
Release: **3.0.0**

The latest release of Xen adds support for Intel's hardware virtualization mechanism, SMP guest systems (with hot-pluggable virtual CPUs), large memory support, trusted platform module support, a port to the IA-64, and initial support for PowerPC architectures as well as numerous bugfixes and minor updates.

RRDTOOL
http://people.ee.ethz.ch/oetiker/webtools/rrdtool/
Release: **1.2.12**

Bugfixes: fewer memory leaks and double-frees, and proper UNKNOWN handling when using N: for updating. New features: RRDtrac on the RRDtool Web site, no more libcgi requirement, and faster graphing.

ZOPE
http://www.zope.org
Release: **3.1.0**

Zope is an application server specializing in content management, intranets, and custom Web applications. It is written in Python and has a large global community of developers and companies.

>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.

ModSecurity

MODSECURITY IS AN OPEN SOURCE WEB APPLICATION FIREWALL

THAT PROVIDES INTRUSION DETECTION AND PREVENTION FOR WEB APPLICATIONS

BY JOHN BUSWELL

Modsecurity is an Open Source Web Application Firewall which provides intrusion detection and prevention for any web application running on the server that Modsecurity is protecting. It is built as a module for Apache, and for the purposes of this article, we will be looking at modsecurity 1.9.1 running on Apache 2.0.55. It works by applying a filter engine to inbound HTTP requests, running the request through a number of built-in checks, user customized filter rules and then, if a positive match occurs, taking a specific action.

SecChrootDir

While the security features Modsecurity provides are excellent, it has one particular feature which I feel makes it the killer application for web security in general. That feature is chroot() capabilities. If you have ever spent the time running ldd against libraries and binaries, manually chrooting Apache (something which becomes far more of a chore once you start adding PHP, MySQL and other extensions into Apache), then you will really appreciate this feature. Once modsecurity is applied to Apache, typically as a DSO module, you simply add the directive SecChrootDir and the path as in the example below:

SecChrootDir /chroot/apache

That's it - you don't need to keep binaries, libraries and other things that you'd normally have to transfer over. Obviously, if you're running CGI scripts, you'll need to make sure those scripts don't require any additional libraries, but as for Apache itself, the job is done. If you are running Apache multi-threaded, then you may need to add LoadFile /lib/libgcc_s.so.1 to get pthread_cancel to work.

File Uploads and Server Identity Masking

Modsecurity has the capability of intercepting files uploaded through POST requests and multipart/form-data encoding or through PUT requests. It also enables you to upload those files to a temporary directory using SecUploadDir and then execute a script through SecUploadApproveScript to authenticate the upload. The modsecurity documentation also provides a good example on how to integrate this feature with the Open Source anti-virus software ClamAV.

While Apache 2.0 supports the ServerToken directive, it still reveals through the HTTPD / HTTPD/1.1 command that you are running Apache. While you can always modify the source, modsecurity provides a SecServerSignature directive which you can use to simply change it to any string that you desire.

Web Application Security Vulnerabilities

Before going into too much detail on the capabilities of modsecurity and how to use them, we will look at the reason you need it in the first place. The Open Web Application Security Project (OWASP) is dedicated to finding and fighting the causes of insecure software. The OWASP community continues to do a great job, and the documentation they provide is an excellent starting point on what to do in order to protect your web server against attacks with modsecurity. It is even more useful if you don't already have event information from an existing Intrusion Detection System such as snort. For the purpose of this article, we will focus on relevant security issues in the top 10 Web Application Security Vulnerabilities published by OWASP:

- Unvalidated Input
- Broken Access Control
- Broken Authentication / Session Management
- Cross Site Scripting
- Buffer Overflows
- Injection Flaws

- Improper Error Handling
- Insecure Storage
- Denial of Service
- Insecure Configuration Management

Four of these issues are problems that can be addressed through other means. We will briefly look at those here and then move on to using modsecurity to protect against the rest. Broken Access Control refers to improperly enforced access control - users having access to information that they shouldn't. This can be addressed through file system enforcement combined with a proper authentication system such as Radius. Apache's access control system can easily be patched to work with Radius.

Likewise, Broken Authentication and Session Management can also be addressed partially that way. Ruby on Rails is a good example of a system that provides an enforceable secure session management system. Session Management is something that you will need to address within your broken application.

Insecure Storage refers to weak (or a lack of) encryption when storing sensitive information such as passwords or credit cards. Typically this information is stored in a database, but the information should be stored in an encrypted manner in case the database is compromised and the data dumped, such as through an SQL Injection attack. Again, this is an application design/coding related issue that needs to be addressed in the program itself.

The last item on the list that we won't cover is Insecure Configuration Management. Modsecurity is an advanced security measure, so if you are looking at modsecurity, we are going to make the assumption that you've already taken the basic steps to secure your standard web server configuration. The Apache documentation is a good place to start, I would also recommend Apache Security by Ivan Ristic and Hardening Apache by Tony Mobily. Both of those books will act as an excellent resource to guide you through securing your Apache configuration.

WHAT TO INSPECT

As with all security inspections, whether its packet filtering, HTTP request inspection or security at the airport, there will always be some kind of performance cost. While the cost of modsecurity is relatively small and the benefits far outweigh the hit to performance, it is possibly to save resources on your web server by inspecting dynamic requests only (PHP, Ruby, etc) and ignoring your static files. This is achieved by SecFilterEngine DynamicOnly.

Modsecurity has the capability of scanning POST requests, and while POST is typically used for uploads, scanning POST requests at least for multipart/form-data and application/x-www-form-urlencoded will enable you to detect potential attacks coming through forms and help protect web applications that may not validate input correctly. Obviously if you don't use POST or forms on your content, using modsecurity to deny these types of requests would be part of your security policy. Scanning specific content types can be done dynamically using environment variables such as MODSEC_NOPOSTBUFFERING.

RULES

Rules are compared to incoming HTTP requests. When a positive match is achieved, the filtering engine will perform the action associated with the rule. Rules are defined with the directive SecFilter followed by a keyword. In its simplest form SecFilter foobar the filtering engine would match any occurrence of foobar in a HTTP request. The power of the SecFilter directive is realized when you use regular expressions instead of just a regular keyword. (Regular expression is a type of tiny programming language designed to match patterns within a block of text.)

While SecFilter is great for performing broad searches, the SecFilterSelective directive allows you to perform accurate pattern matching within specific locations of a HTTP request. The format is **SecFilterSelective LOCATION KEYWORD [ACTIONS].** The SecFilterSelective command supports all CGI variables as well as a long list of location identifiers. These identifiers are listed here in the modsecurity manual.

Modsecurity supports inverted selection using the exclamation mark (!) in front of a particular keyword or location. Modsecurity also supports cookie processing, by default cookies use version 0

(Netscape-style) but it also supports RFC 2965 version 1 cookies.

ACTIONS

Actions tell modsecurity what to do when a specific match occurs. Actions can be combined within quotes and comma separated, for example SecFilterDefaultAction "deny,log,status:500" will deny, log and return a 500 status back to the client. A default action should be defined as a catch all, similar to that in packet filtering. Depending on your security policy, you might want to the catch all to allow and log, or deny and log. Actions can be defined per rule at the end of a SecFilter or SecFilterSelective directive. It is also possible to create lists which fall under the same set of actions. This is achieved by prefixing a SecFilterSignatureAction directive with the action at the top of a list of SecFilter / SecFilterSelective directives. Those filter commands below the SecFilterSignatureAction will inherit that action instead of the default.

The pass action allows a filter to match and perhaps to log a specific event, before allowing filtering to continue. The allow action is a positive match, filtering stops and the request is permitted to continue. Likewise, the deny action will stop filtering and prevent the request from continuing. If a particular request is denied, you can use the status: action to respond back to the client with a specific HTTP result code. On a filter match, the redirect action enables the server to redirect to a given URL. If you have mod_proxy installed, the proxy: directive can be used to rewrite the request through the internal reverse proxy, for web content acceleration and other applications.

The exec action enables you to execute an external application handler on a filter match, perhaps to mitigate a specific attack. The skipnext and chain actions allow you to manage multiple rules as part of a group, while the pause action allows for a specific delay before responding to a request. There are log, nolog, auditlog and noauditlog actions to dictate how a specific request may or may not be logged. There are also actions for custom logging – id, rev, msg and severity. The mandatory action marks a rule, chain or rules for mandatory inheritance in subcontexts. The setenv and setnote actions provide the ability to set or unset a named environment variable on an Apache environment variable.

LOGGING AND AUDITING

SecFilterDebugLog allows you to configure a debug log, along with the SecFilterDebugLevel directive, which defines a 0-9 scale for debugging level, 0 being none and 3 being most detailed, with the rest used for internal debugging. Modsecurity, however provides a detailed auditlog feature which is far superior to the typical Apache log when trying to trace back the activities of a user or an attacker. This feature can generate a lot of log data as each request contains full HTTP headers, so if you have a busy server, expect a lot of logs. The SecAuditEngine On/Off directive toggles the feature; it also supports RelevantOnly and DynamicOrRelevant which logs only Relevant and Dynamic requests respectively. The SecAuditLog directive defines the log.

When using ModSecurity on dynamic requests, you should change any AddType application/x-httpd-php .php lines in your config to AddHandler instead of AddType so that Apache handles the requests in a manner which Modsecurity can audit properly. The change has no effect on functionality, it simply enables modsecurity to utilize the Apache internal handler.

UNVALIDATED INPUT

The top security problem on the OWASP list is processing input data without properly validating it. While this is really something the application developer should have taken into account, in practice not all web application developers take security into account. The combination of regular expressions (regex) and the SecFilter command enable the administrator to configure filter rules to look for any additional data, such as paths or escape code sequences, tacked onto the end of valid input.

CROSS SITE SCRIPTING

A cross site scripting attack (XSS) occurs when HTML and/or Javascript code is injected into a web page by an attacker and that code is then executed by

other users who view the page. When successfully executed, an attacker could obtain access to the cookie within a session and thus gain full control of your web application. This type of an attack is filtered out by using SecFilter "<script" and SecFilter "<.+" which will prevent both Javascript and HTML code respectively from being injected.

Many applications such as a CMS, forums and so forth which actually want HTML in parameters. In such cases, you can use SecFilterSelective within VirtualHost or Location directives in Apache to permit and control the exact parameters that need to be permitted to the web applications.

BUFFER OVERFLOWS

Buffer overflows involve overflowing the stack and adding assembler code in an attempt to get that code executed. Modsecurity enables you to prevent such attacks by using SecFilterByteRange 32 126. This permits ASCII code between decimal 32 (SPACE) and decimal 126 (~) (see man ascii on any Linux system for information). However the character encoding could prevent this from working all the time, so to backup this command you can use **SecFilterSelective THE_REQUEST "!^[\x0a\x0d\x20-\x7f]+$"** which will achieve the same thing with regular expressions.

INJECTION FLAWS

SQL injection and operating system command execution are two common types of injection flaws that plague web applications. SQL injection involves placing SQL commands into a request which, if the application is not carefully coded to protect its database, will result in those SQL commands being executed against the database. This could be easily used to dump user information or credit cards or simply to delete tables from the database. To protect against these types of attacks, the following will check for most SQL attacks by checking for SQL commands within the request:

SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFitler "select.+from"

A similar technique, such as filtering for "bin/" or "opt/" within ARGS or other requests will prevent most attacks. However, if you have executables accessible from the web server, covering those paths is necessary as well.

IMPROPER ERROR HANDLING

Improper Error Handling refers to the display of error messages and internal information to the user in the browser. If a malicious user can reproduce these types of errors, they can learn more about the system in order to develop an attack. In order to protect against this type of error, you must enable Output filtering, which is only supported in the Apache 2 version of modsecurity. The SecFilterScanOutput On directive will enable the feature, then simply use OUTPUT as the location for SecFilterSelective. This makes it easy to catch output errors in languages such as PHP, which can be stopped with :

SecFilterSelective OUTPUT "Fatal error:" deny,status:500
ErrorDocument 500 /php-fatal-error.html

DENIAL OF SERVICE

Modsecurity 1.9 combined with a tool called http-guardian (*http://www.apachesecurity.net/tools*) can be used to provide the stateful information required to defend against Denial of Service (DoS) attacks. The SecGuardianLog command sends all access data to another program using the piped logging feature. The http-guardian tool uses a blacklist tool to interact with iptables in order to dynamically block offending IP addresses during a DoS attack.

TESTING

Modsecurity comes with a small test utility called run-test.pl which enables you to send to a host HTTP requests contained within a specific file. The utility allows you to craft HTTP requests that might be used by an attacker or a valid user in order to make sure your filter rules and actions are working correctly. Nikto is an Open Source web server scanner which can perform comprehensive tests against web servers and is definitely worth a look. The OWASP project is also a good source of tools and white papers on

building custom security assessment tools for web applications.

PERFORMANCE AND CONCLUSION

Modsecurity has a small performance cost, and the time to process regex is extremely short. Our testing showed around a thousand rules on Apache 2 running on a server with a 2.4 Ghz processor took under 6 ms. With the latest release of modsecurity for Apache 2 there are a number of performance measurements which you can inject into the CustomLog directive in Apache 2. These are specific to modsecurity 1.9 and above. The directives are mod_security-time1, mod_security-time2 and mod_security-time3. Time1 represents that modsecurity initialization is complete and the body of the request has been read if POST scanning is enabled, Time2 the rule processing has completed, and at time3 the response is ready and is waiting to be sent to the client. The output is displayed in microseconds.

Overall, modsecurity is extremely easy to build and install and with a good security policy design, is relatively simple to configure. If packet filtering on the edge of your network is your first line of defense, then Modsecurity should be part of the last line of defense on your server. Modsecurity prevents programming errors and security flaws with third party applications from becoming catastrophic security problems. If you are running Apache, and not running Modsecurity, you need to look carefully at deploying Modsecurity sooner rather than later.

John Buswell is co-founder and Chief Technology Office of Spliced Networks LLC. He can be reached by email (johnb@splicednetworks.com).

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

Asynchronous JavaScript and XML (AJAX)

AJAX IS A COMBINATION OF DIFFERENT TECHNOLOGIES TO PROVIDE A FRAMEWORK FOR BUILDING INTERACTIVE WEB APPLICATIONS

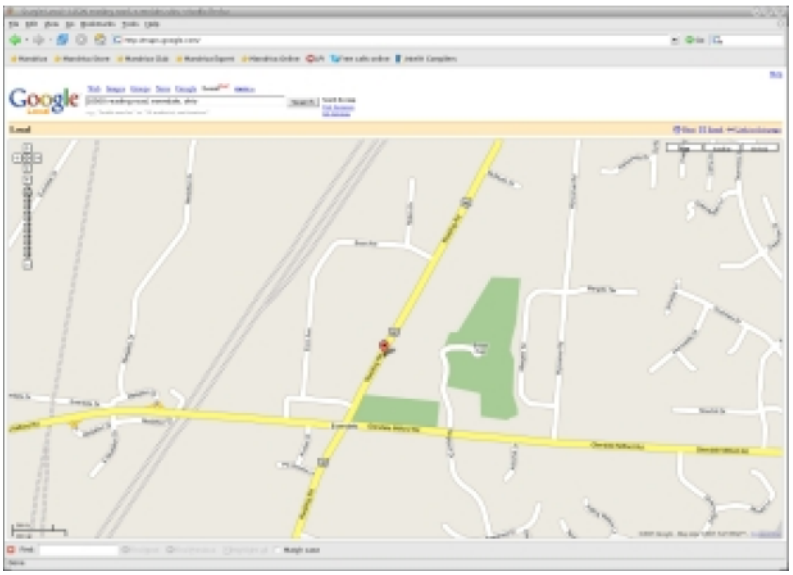
BY ABUL ASIM M.R. QARSHI

Asynchronous JavaScript and XML (AJAX) is a new development approach for building more rich, interactive and responsive web applications. AJAX is not a technology in itself, but rather is a way of using several technologies including HTML or XHTML, Cascading Style Sheets, JavaScript, the Document Object Model, XML, XSLT and The XMLHttpRequest Object.

The interaction mechanism of a web application is different from your average desktop application. Each instance of a web page needs to communicate with the server in order to get the response which it needs to update. This is time consuming and lowers the user experience.

Let's consider an application which displays photos as an example. Since the application is running on the web server and not at the client's side, when the user wants to see the next photo, the whole page has to be rendered from scratch even though 95-99% of the content never changes. Now consider this same application again, but this time running as a desktop application - when the user clicks to see the next photo, it goes smoothly and only has to render the photo, because it's running entirely on the client side.

The gap that exists between the perceived behavior of desktop and web applications is closed down by AJAX since it is processed at the client's side. AJAX can be used to make Rich Internet Applications (RIA) which can have an interface consisting of a desktop-like GUI component running on a standard browser without increasing the size of the document.



HTTP://MAPS.GOOGLE.COM (MAINSTREAM AJAX)

HOW AJAX WORKS

AJAX Applications use an AJAX Engine which resides in an intermediate application layer between the user and web server. This AJAX Engine is written purely in JavaScript and some times placed in a hidden frame. Although some people might argue that using an intermediate layer will make it less responsive, the opposite is true in the case of AJAX since the resulting applications are actually more responsive.

How is this possible? When web page is accessed by the user for the first time, the AJAX Engine is loaded by the browser. This engine is responsible for rendering the user interface as well as fetching data from the web server in form of XML by using the XMLHttpRequest object. Now the whole application is running on the AJAX Engine and doesn't need to render the page at the server. The AJAX engine allows the user's interaction with the application to happen asynchronously (independent of communication with the server). This means that the user is never staring at a blank browser window while waiting around for the server to do something.

AJAX DEVELOPMENT TOOLKITS

AJAX Engines are rather complex pieces of code written in javascript, and it's not easy to write one for yourself. Fortunately, there are several third party Development Toolkits to write AJAX based web applications. The three listed below are by no means an exhaustive list, but they are a nice place to start.

- **Bindows:** *http://www.bindows.com*
- **Dojotoolkit:** *http://dojotoolkit.org*
- **Sajax:** *http://www.modernmethod.com/sajax*

AJAX BASED WEB APPLICATIONS

As AJAX has gained in popularity, it's gotten out of the laboratries and onto production servers in the form of both simple and complex real world web applications. It even seems to have become one of the favorite technologies at Google since they have been creating so many applications using AJAX. Below is a short list of some things that AJAX is currently being used for out in the real world.

GOOGLE MAPS

http://maps.google.com

One of the best examples of the capabilities of an AJAX application is Google Maps. For those of you unfamiliar with this application, it is a quick loading, responsive map of the world which offers a great deal of functionality including the ability to search for locations and directions. The really impressive thing, though, is the fact that it responds basically in real time to both the searches and to keyboard and mouse commands to move across the map and to zoom.

We're not talking about zooming into a simple wire-frame style map either. Google Maps allows for browsing a standard style map complete with borders, cities, etc, a satellite map, or an overlay of the standard map on the satellite image. It really has to be one of the most impressive AJAX applications that I've seen.

GMAIL

http://www.gmail.com

Google's web mail service

MEEBO

http://www.meebo.com

A multiple client instant messenger built with AJAX which allows you to use AIM, Yahoo!, MSN, ICQ and Jabber/Gtalk without having to install any software.

MIDNIGHTCODERS

http://www.themidnightcoders.com/examples/

Several AJAX based application examples are listed here.

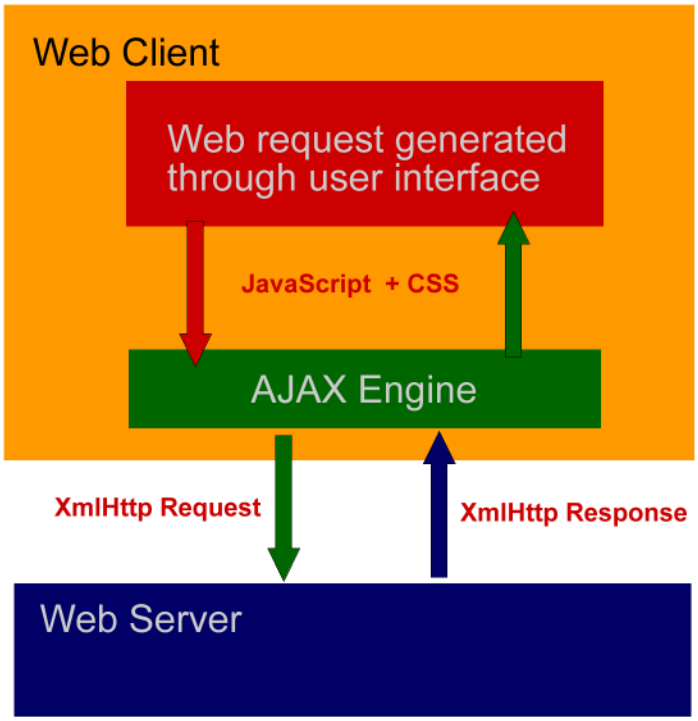
CALENDAR HUB

http://www.calendarhub.com

An online calendar which can either be kept private or made public.

Now that we have done a brief overview of AJAX and you've had a chance to see several services based on the technology, we hope that you consider its use in the future for your web applications. In fact, there are several rapid web development frameworks, including Ruby on Rails, which allow you to make use of AJAX easily.

ABUL ASUM M.R QARSHI IS A NETWORK SECURITY SPECIALIST FOR SPLICED NETWORKS LLC BASED OUT OF PAKISTAN. ABUL HAS A PASSION FOR NETWORK SECURITY AND BLEEDING EDGE WEB BASED TECHNOLOGIES. ABUL CAN BE REACHED VIA EMAIL (AQARSHI@SPLICEDNETWORKS.COM).



MNX SOLUTIONS

Technology Solutions for Your Business Problems.

SAP and Linux
Flexible. Innovative. Scalable.
Considering SAP on Linux? It will save you money. We can show you how.

LINUX Solutions
Open. Experienced. Certified.
Looking for a Linux solution? Let us design a solution that will meet you business needs.

SAN Solutions
Performance. Security. Consolidation.
Realize high performance and availability with our open systems storage solutions.

High Availability
Cost Effective. Manageable. Reliable.
Linux Highly Available Solutions. We know Linux and High Availability. Let us help you design and implement your solution.

info@mnxsolutions.com http://www.mnxsolutions.com (888) 877-7118

Keeping web projects on track and on budget with Ruby on Rails

RUBY ON RAILS IS A RAPID WEB APPLICATION DEVELOPMENT FRAMEWORK

LEARN HOW RAILS CAN HELP YOU KEEP YOUR WEB PROJECTS ON TRACK AND ON BUDGET WITH RECORD DELIVERY TIMES

BY JOHN BUSWELL

Ruby on Rails is an open Source web framework that is optimized for rapid web application development. Rails provides a structured framework that makes development feel natural and easy to maintain. Ruby is a relatively easy programming language to learn; any programmer with even just a vague idea of Java, Python or another object orientated programming language will pickup Ruby fairly quickly.

With a little effort, Ruby on Rails will allow you to quickly develop and modify web applications in an efficient and cost-effective manner. Projects typically go over budget due to unrealistic time frames, unanticipated problems or the varying skills of developers on the team. Ruby on Rails will help mitigate these traditional problems by providing a fast and easy to follow framework for building web applications. Ruby on Rails applications are database centric, and Rails provides an object-relational management layer called ActiveRecord that significantly reduces the headaches caused trying to map object based programming languages with data contained in relational databases.

Rails provides instant gratification - you make a change, you point your browser, and you see the change in effect. It's instant. This has a key benefit when demonstrating an application or proof of concept application to a client. A lot of problems around customer web applications center around the ability of the engineering team to communicate with the customer and many companies have a management “translator” who interacts between the customer and the engineers.

Rails eliminates this need, as the customer can see the application in real time. If the developer misinterpreted the customer's feature request, the developer can quickly load up an editor, modify the feature and show the customer the change instantly. This allows for faster and easier communication with the customer and will increase the customer's atisfaction with your business. In fact, many non-technical customers who have dealt with developers using traditional means, such as the document, develop, demonstrate, document, develop, and demonstrate cycles, will be highly impressed by the instant nature of rails. I had one businesswomen refer to it as “magic”.

If you are still not convinced about the speed and simplicity of Rails, then I suggest you take a look at the screencasts that are available over at *http://www.rubyonrails.com/screencasts*.

GETTING STARTED

RUBY

Ruby is an interpreted, high-level object orientated programming language, and in some situations it may not perform as fast as lower-level languages such as C. However, there are a number of things you can do to improve performance. Slow running programs typically have a few locations where the processes are heavily hit. You can use the Benchmark module and code profiler that come with Ruby to help locate and fix these types of problems. Since most web developers write code these days in high level languages such as Python, Perl and Java anyway, and Ruby stacks up very well in performance compared to these languages, so the performance hit, if any for a specific application is well worth the cost and time saving benefits of the language. If you would like to learn more about the Ruby Language itself you can look at *http://www.ruby-lang.org/*. If you would prefer a book, I would highly recommend Programming Ruby: The Pragmatic Programmer's Guide by Dave Thomas.

RUBYGEMS

RubyGems is the package management system for Ruby. The command was shortened to gem. Gems interacts over the Internet with rubyforge.com, a huge repository of software code maintained by the rapidly growing Ruby community. RubyForge provides

access to over 1,100 hosted projects, so you will often find something you are looking for without having to code it yourself.

BUILDING THE ENVIRONMENT

Rails can be integrated with Apache, Lighttpd and many other web servers that support SCGI or FastCGI. Lighttpd is a good option, and was reviewed in last month's issue of O3. To get started you will need Ruby and RubyGems. The recommend builds are listed on the Rails download page (*http://www.rubyonrails.com/down*).

BUILDING RUBY

The recommended release for Rails is currently 1.8.2. Untar it, and go through the usual POSIX motions (./configure && make && make install). If you want to make use of the ruby documentation tool (ri) then you will also want to run make install-doc.

BUILDING RUBYGEMS

Installing RubyGems is trivial - simply untar and run ruby setup.rb from the rubygems-0.8.11 directory.

INSTALLING RAILS

Now that you have both Ruby and RubyGems installed, you can use the gem command to grab rails. You will need to make sure that you have Internet access on the system you are using Ruby on before running the following command:

gem install rails –include-dependencies

As simple as that, rails is now on your system. Keep in mind that Rails is database centric, meaning you will need to have some kind of database available to you in order to utilize Rails. A wide range of databases are supported including MySQL and PostgreSQL. For this article, I used PostgreSQL 8.1.

ARCHITECTURE

The Rails architecture is built around MVC, ActiveRecord (Object-Relational database management) and URL mapping. These are core concepts for using Rails so we shall look at them briefly.

MVC

MVC is an architecture designed by Trygve Reenskaug back in 1979. It is a Model, View, Controller architecture. The model manages the state of the application. It is not just a representation of data since it enforces the business rules that are applicable to that data. Such rules might be that shipping within the state should always be UPS Ground because it will get there overnight regardless, that sales tax is charged based on the destination within the state or that customers outside the European Union are not charged VAT.

The view is the visual presentation of the data to the user. This is the user interface, and from a rails point of view, is usually the HTML that is displayed to the user's browser. The view might be different for different perspectives, so the administrator might see all the users, while a single normal user may only see their preferences.

As the name suggests, a controller maintains control of the application. It takes events from the user, interacts with the model (data) and provides a new view to the user.

ACTIVERECORD

Anyone familiar with injecting SQL commands into their PHP or C code is engaged in whats called Database-centric programming. If you know SQL, then this is a relatively easy and pain free method of obtaining data from a database quickly and easily. If you are a good programmer, you probably write centralized functions that are called throughout your code, so that if you need to modify how your application interacts with the database or to change databases, you have a relatively pain free method of doing so. Unfortunately, not everyone is a good programmer, and maintaining code over time with hundreds of SQL statements throughout thousands of lines of code can become a very painful task.

Rails utilizes a method called Object/Relational Mapping where database tables are mapped to object classes. This is not an easy thing to achieve and often requires considerable amounts of XML configuration files. ActiveRecord solves this problem by providing an ORM layer within Rails. ActiveRecord, like many aspects of Rails, relies on convention and sensible defaults, making it easy to modify and customize to your specific needs. ActiveRecord integrates seamlessly with the rest of the Rails framework.

url mapping

The last important concept to grasp is that Rails utilizes URLs to map requests to a specific controller and action.

http://192.168.99.202/rails/helloworld/greet/hello

In the URL above the *http://192.168.99.202/rails/helloworld/* identifies the application, the *greet/* selects the controller (greet) and the *hello* provides the action to invoke. However, Rails is reasonably flexible about how it parses the URLs, so it is possible to parse them differently should you wish to do so.

BUILDING A SIMPLE APPLICATION

We will now walk you through the creation of a simple "hello world" application in rails. The steps we will take will be to create the application, add a controller, add a view, add some dynamic content and then test the application.

CREATING THE APPLICATION

Creating applications with rails is trivial. With rails installed earlier, we simply change directory over to where we want to store our new project then run rails <project>, for our application we will run rails helloworld. This will create a new directory called helloworld, and populate it with the default files for rails. To start the test server (assuming your not running it via FastCGI under another web server) simply run helloworld/scripts/server and then go open up a new terminal.

ADDING A CONTROLLER

Next we will need to add a controller. We will add one called Greet as it will manage the greeting to the user.

```
ruby script/generate controller Greet
exists app/controllers/
exists app/helpers/
create app/views/greet
exists test/functional/
create app/controllers/greet_controller.rb
create test/functional/greet_controller_test.rb
create app/helpers/greet_helper.rb
```

Now we edit apps/controllers/greet_controller.rb. As you can see, it already has some code in there for us:

```
class GreetController < ApplicationController

end
```

We will add the action "hello" to our controller:

```
class GreetController < ApplicationController

  def hello

  end

end
```

If you point the browser at the server http://192.168.99.202:3000/greet/hello/, you will get a message about a missing template. This is produced because we haven't added the view yet. The view is added under app/views/greet/<action>.rhtml. In our case hello.rhtml.

CREATING THE VIEW

Adding the view is simple. We just create hello.rhtml with some html in it:

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<b>Hello World!</b>
</body>
</html>
```

Point the browser at Rails again, and now we get Hello World!

ADDING SOME DYNAMIC CONTENT

Just to demonstrate how easy it is to add dynamic content in Rails we will modify our code now to display the time. To do this, we need to update the controller (greet_controller.rb) to capture the time by modifying the hello action to:

```
def hello

@time = Time.now

end
```

Then you simply edit the hello.rhtml file to display the time:

```
<body>

<b>Hello World!</b><p />

The time is now <%= @time %>.

</body>
```

It's as simple as that. We now get the following in our browser window:

Hello World!

The time is now Mon Dec 19 12:57:43 EST 2005.

CONCLUSION

Overall, Ruby is an intuitive and easy language to learn and Ruby on Rails provides a fast and optimized framework for developing web applications quickly. Not only is it great for prototyping and proof of concept applications, but you can simply expand upon the original proof of concept code and utilize it as a base for the actual application. Ruby on Rails is well worth a look whether you're looking for a simple in-house project or trying to keep unrealistic project deadlines with traditional web solutions. The bottom line for businesses, especially web development businesses, is that Ruby can save you time and resources regardless of the size of the project.

Performance Hosting
quality service

Whether you are promoting a new business venture or expanding your existing one, you need a solid and secure hosting partner.

Why settle for less?

BLACKNIGHT SOLUTIONS

www.blacknight.ie sales@blacknight.ie +353 (0)59 9137101

Rapid Web Development

RAPID WEB DEVELOPMENT CAN HELP DEVELOPERS PRODUCE PROOF OF CONCEPT AND DEMO APPLICATIONS IN A MATTER OF HOURS AND PERMIT RAPID APPLICATION CHANGES BASED ON CUSTOMER FEEDBACK

BY JAMES HOLLINGSHEAD

Once upon a time, having a web page meant slapping together some HTML and uploading it to a server where it could be accessed. It didn't really do much. In fact, most web pages tended to look like electronic versions of information booklets.

These pages were informative and they were easy to make. However, they were boring and not very useful past imparting simple information.

Then people started coming up with the idea to start doing things other than just imparting information on the Internet They thought it would be great if we could buy things online without having to go to the nearest store or that we might want Internet-based services (like ways to find other pages).

Our quick, simple HTML world got a lot more complicated really quickly. After a lot of trial-and-error, several security fiascoes, and much gnashing of teeth at pages that really didn't do what people thought they should, things started coming together. This has lead to the rise of rapid web development.

WHY SHOULD I USE RAPID WEB DEVELOPMENT?

First and foremost, rapid web development saves you time. This does one of two things for you – if you run a business which creates web sites and web apps for clients, it allows you to take on more jobs since the overall time for every job is reduced. If you have developers working on internal applications like the non-profit I worked at, it lets them take care of their projects more quickly so they can do other useful things, thereby saving you money that you might otherwise have to spend on additional development staff.

Second, rapid web development technologies allow you to easily create maintainable code. This is a wonderful thing if you have a new developer taking over a project since it will take less time for them to understand what the code is doing. It's also nice because it is often quite a long time between when the application is developed and when it is updated. This can save hours that your developers would otherwise spend in the pursuit of pulling out their hair because they suddenly find themselves looking at something they haven't even had to think about for months. I speak from experience in this. Maintenance is a developer's worst nightmare and anything that can make that part of their life easier will lower their blood pressure a few points.

Third, flawed, buggy software costs your business money. It costs you the time of your developers to fix security flaws and keeps them from working on things that make money for your company. Releasing flawed software can also damage your reputation. Let's face it – your greatest means of getting new business is from current clients. Advertising is a great way to get eyes looking at your company, but if they start looking around and find that a lot of people are unhappy with your services, they will look somewhere else. Rapid web development helps here as well since most of the technologies are designed to be secure from the ground up instead of leaving security as an afterthought or leaving it up to the developers to do all of the security heavy lifting.

Finally, it's really pretty easy to pick up. Ruby, one of the languages discussed below, can be learned in a couple of days and the Rails framework that uses it can be picked up easily as well. You can even port your existing applications pretty easily if you so desire.

Now that all of the reasons to use rapid web development technologies have gotten your attention, let's take a look at the concept that the technologies are based on.

MODEL VIEW CONTROLLER

At the core of the rapid web development technologies that we'll be discussing in this article is the Model-View-Controller (MVC) concept. While not a new idea (it was first described in 1979), it has found its way into many peoples' vocabulary for the first time.

Basically, what it boils down to is that the software's architecture is broken down into three parts – the Model (data model), the View (user interface), and the Controller (control logic). This allows modifications to be made to one area of the software without impacting the other two a great deal. That means that you can change the way that the end result looks to your customer (user interface) without having to change either the data model or the control logic behind the application. The same is true for changing the business logic (data model) or what happens when you click a button (control logic).

This makes maintenance easier because your developers don't have to look for things that may break in the other two areas if they need to make a change to any part of the program. It also means that the application is more secure because none of the business or control logic is being presented to your customers or anyone who might want to exploit your application (which was a big problem with previous approaches to web development).

While MVC is a nice, and rather nebulous concept, what we're really interested in is the actual implementation of rapid web development and how it can help us get our business done efficiently and help save us money by saving us time. That said, let's take a look at some of the more popular frameworks that let us quickly and safely get our programs out of the concept phase and onto the server where they can be used.

RUBY ON RAILS

Rails is a rapid development framework based on the Ruby language. Ruby is a language that started out being roughly modeled on Perl, so it's also used to rapidly develop programs that aren't used online.

At the heart of Ruby is the idea that you shouldn't have to configure everything because the normal, everyday things behave exactly the way any sane person would expect them to. This makes development go much faster because your programmers don't have to worry about every tiny detail in order to create the application.

Rails builds on this by giving you the most common pieces of basically any web application that you could care to create. It includes pre-written libraries for things like communicating with databases, data validation from forms, sending and receiving email, formatting date and time information, and interactive client-side functionality with AJAX. In fact, all of this is set up for your new application with one command. After that, you just start filling in the blanks.

It has a fairly large amount of documentation online and a helpful and supportive community. In fact, the first edition of one book on Ruby, Programming Ruby (currently in its second edition), has been made available online by its authors at (*http://www.rubycentral.com/book/*)

While all of the things stated above are great reasons to use Rails, one of the best reasons is the fact that you don't have to recompile the program in order to cause any updates to your web application to take effect. That's right – you can sit there with your client and make changes on the fly while they give input without having to wait for the program to recompile after every change. Anyone who has ever had to go get a cup of coffee and then still had time left over while their program compiles can tell you how much this can decrease the development time.

PHP SMARTY

Smarty (*http://smarty.php.net/*) is based on PHP, which has been one of the most used languages to create web applications. Like PHP, it has a large number of plugins and has a built-in debugging console. It provides caching for all or just parts of a page and also supports the use of configuration files to keep common values in one location, allowing a change in one place to effect the entire program.

For the presentation portion of the framework, Smarty uses special tags that have a syntax fairly close to normal HTML. While these templates, which closely resemble what the resulting page will look like, contain no PHP themselves, they are compiled into PHP code in order to reduce the time that the server spends parsing the code.

Since PHP has been around for a while, there is quite a lot of documentation on the base language and the documentation for Smarty is available online under the Documents section of the Smarty homepage.

JAVA STRUTS

Struts (*http://struts.apache.org*) has been around the longest of any of the frameworks that we're covering here. Having been created by the Apache project as a way to tie together things like Java Server Pages,

servlets, custom tags, and other resources into a unified framework, it supports industry standards so it's compatible with other Java technologies.

Struts is also very mature and has a huge amount of documentation available online (not to mention more paper based books than I care to count). The major downside of Struts that most people point out is the fact that, being based on Java, it tends to be rather verbose, so unless you have a good editor which allows you to autocomplete things like variable names, it takes a while to develop in just because of the amount of typing. On the upside, it is very powerful and has vast amounts of libraries which provide a lot of the functionality that most online programs use.

TURBO GEARS

Turbo Gears (*http://www.turbogears.org*) is a relatively new framework based on the Python language that provides a four-tier approache to rapid web development. Being new, there are still a lot of features being added and new documentation being made while they reach a stable release.

Having said that, it looks like an interesting project. SQLObject makes database queries look more like an object oriented programming language (think C++ or Java). CherryPy is used to quickly create dynamic content. Kid provides an XML templating system. Finally, MochiKit is a JavaScript library that allows programmers to work with AJAX capabilities.

CONCLUSION

Whether it's based on Ruby or Java, PHP or Python, the use of rapid web development technologies can help decrease the time that you're waiting for your web-based applications to make it to the production server while keeping the problems with maintenance and security to a minimum.

The next time you need to start a new web application, give some thought to how rapid web development technologies can help you go live sooner and be sure to check the other articles in this issue of O3 which deal with Ruby on Rails in a more in-depth fashion.

James Hollingshead is the Executive Editor for O3 Magazine. James can be reached via email (james@o3magazine.com).

SCTP vs TCP

SCTP IS A STREAM CONTROL TRANSMISSION PROTOCOL DESIGNED TO SIT ON TOP OF IP

IT OFFERS ACKNOWLEDGED ERROR-FREE NON-DUPLICATED TRANSFER OF DATAGRAMS

BY RAJA HAMMAD

TCP has been the most dominant and successful connection-orientated transport layer protocol along with its connectionless counterpart, UDP, for the last few decades. During these years, TCP has gone through many changes at the design level, such as advancements in its congestion control mechanisms, that kept it reliable for data transfers over connectionless IP protocol.

The Internet has grown explosively in the last decade and new applications have emerged that have requirements not offered by TCP and UDP. One such application is the transport of Public Switched Telephone Networks (PSTN) signaling messages over IP networks which has stringent requirements in terms of reliability and timing. The limitations of TCP, as discussed in RFC 2960, are:

- TCP offers reliable data transfer service only and transmits data in a strict order. This may be useful for many applications but other applications, such as telephony signaling, require only reliability and partial ordering of data.

- A TCP stream is a sequence of segments and strict order-of-transmission requires delivery of those segments in a particular order. This can potentially introduce delays in data delivery, a problem known as **head-of-line blocking**. This happens when a single TCP segment is lost which results in blocking the whole data until the lost segment is recovered. Such delays are not acceptable in applications such as telephony signaling.

- TCP is a stream-oriented protocol which means it does not define the message boundaries. Applications must define their own record marking.

- Multihoming is not supported in TCP, which is required for high availability links in applications such as telephony signaling.

- Security is an important feature, required for many applications but TCP is known to be relatively vulnerable against SYN flooding attacks.

UDP is also not suitable for such applications. It offers unreliable, unordered data services and has no built-in congestion control mechanisms. For example, applications requiring reliable delivery and partial ordering cannot take advantage of UDP and must add their own reliability mechanisms on top of UDP.

SCTP was initially developed by the Signaling Transport (SIGTRAN) working group in IETF in order to overcome the limitations imposed by both by TCP and UDP for telephony signaling applications and later it was standardized for broader range of applications. In the following sections, I will be discussing the new features of SCTP that make it an improvement over TCP and UDP.

SCTP stands for Stream Control Transmission Protocol, a new transport layer protocol. SCTP is designed by keeping features from TCP and UDP in addition to its own characteristics. Like TCP, it is a unicast protocol with reliable transmission, congestion control and avoidance features. Like UDP, it is message oriented and supports unordered data delivery. The following section briefly outlines two prominent features of SCTP along with congestion control handling.

MULTIHOMING

Multihoming provides network redundancy and high availability for a multihomed host by setting up multiple IP addresses. Unlike TCP, SCTP allows multihoming by setting up an association (communication relationship between two hosts) between two hosts (either or both hosts can be multihomed), thus providing path redundancy between the two endpoints.An SCTP host sends data

PHPAudit

Protect Your Investment. Choose PHPAudit.

- License your software
- Disable licenses in realtime
- Distribute your software
- Customer Management
- Billing and Invoicing
- Central Administration area
- Host unlimited order systems
- Complete client Login Area
- Full featured shopping cart
- Supports popular pay gateways
- Robust remote API
- In business and stable for 3+ years

Product Editor Shown Above

PHPAudit is web based software for Independent Software Vendors & Web Developers.

PHPAudit Lite 15 FREE Edition

- Up to 15 Active Licenses
- No Ads or Hidden Costs
- Total Cost? $0.00

PHPAudit Owned Version

- Starting at $9.99 for 12 Months
- Starting at $99.00 One Time
- 1 Year Free Upgrades & Support

PHPAudit Hosted Version

- Starting at $4.99 per Month
- ionCube Encoder Access Available
- Lifetime Free Upgrades & Support

PHPAudit + ionCube Bundles

- Starting at $234.99 One Time
- Full Standard Support for ionCube
- 1 Year Free Upgrades & Support

Visit us on the web at http://www.PHPAudit.com

Use this coupon until Jan. 31, 2005 for 10% off any order: O3Magazine

on the *primary transport address* (the sender selects one of the multiple addresses of the receiver as primary transport address) and provides a mechanism for the sender to monitor the reachability of the backup address(es). Thus, in the case of failure of the primary address, SCTP can transparently switch over to the backup address without application layer intervention. The primary address is selected during the initiation of an association. However, this can be changed later by the user application. It is to be noted that path redundancy is not recommended by the current standard for load sharing purposes. However, work is in progress to extend the capabilities of SCTP for load sharing as of Internet-draft *Load Sharing in SCTP.*

MULTISTREAMING

This feature in SCTP allows data to be transmitted in multiple streams within a single association. This allows data delivery independent of data transmission mechanisms within SCTP association; message ordering is preserved within a stream in an association whereas data loss detection, retransmission timer control, etc. are maintained within the whole association. This is potentially useful for many applications where two or more independent sequences of messages can be delivered without inter-dependencies.

For example, a web page consists of different independent multimedia objects e.g. Java applets, images, etc. and they can be delivered in different streams within a single SCTP association. For instance, if an image is lost on its way to the receiver, this will not affect the delivery of the Java applet. This is in contrast to a TCP stream where the Java applet will not be delivered to the user application until the lost segment of the image is retrieved.

CONGESTION CONTROL

SCTP's congestion control algorithm behaves similarly to TCP's well proven rate-adaptive windows-based congestion control scheme. This ensures that, in case of network congestion, SCTP will adjust the packet sending rate accordingly. Moreover, SCTP provides reliable transmission, retransmission of lost packets and detection of reordered, duplicate and corrupted packets. However, SCTP congestion control scheme differs from TCP in many ways.

The following section compares and contrasts the major features between TCP and SCTP: [Note: One important thing to keep in mind is that all streams within a single SCTP association are subjected to a common flow and congestion control mechanisms.

- Like TCP, SCTP uses a slow start and congestion control avoidance scheme to gradually increase the sending rate and transitions from slow start to congestion control avoidance phase to avoid congestion collapse in the network.

- Like TCP, SCTP uses three variables - receiver advertised window (rwnd) , congestion control window (cwnd) and slow-start threshold (ssthresh) in order to control the transmission rate. However, SCTP requires one additional variable, partial bytes acknowledged, to calculate congestion control window growth.

- SCTP has a fast retransmit algorithm based on Selective Acknowledgment (SACK) akin to TCP. However, unlike TCP, SCTP does not have an explicit fast recovery phase, but rather, this behavior is incorporated in fast retransmission by using SACK.

- The use of SACK is mandatory in SCTP as opposed to TCP where it is an optional feature.

- SCTP is designed to support multihomed hosts. This feature complicates the SCTP congestion control process and requires a separate congestion control parameter to be set for each of the destination addresses in the association.

- The multistreaming feature of SCTP allows it to deliver data to upper layers, even if some data chunks are missing, as long as those missing chunks are not part of a single stream. This can affect cwnd calculation.

The following table summarizes the similarities and differences between SCTP and TCP. One of the noticeable difference is that of the way SCTP establish an association. SCTP association is established by exchanging at least four packets (INIT, INIT-ACK, COOKIE, COOKIE-ECHO) as opposed to TCP. This may seem a little excessive from an

overhead standpoint, but interestingly, SCTP allows data exchange with two of the packets, COOKIE and COOKIE-ECHO, without compromising on the security.

Another prominent feature of SCTP is to allow data ordering as an optional feature. This means that ordering can either be preserved within the stream or data can be delivered without any order. Moreover, SCTP allows partial reliability by which reliability can defined on per message basis, allowing reliable and unreliable messages to be multiplexed over a single association. This can be particularly useful for real-time applications such as VOIP, video streaming, etc.

Feature	SCTP	TCP
Connection-oriented	yes	Yes
Connection establishment	Four-way	Three-way
Full duplex transmission	Yes	Yes
Reliable data transfer	Yes	Yes
Partial reliable data transfer	Optional	No
Ordered data delivery	Yes	Yes
Unordered data delivery	Yes	No
Flow and congestion control	Yes	Yes
ECN capable	Yes	Yes
SACKs	Yes	Optional
Message boundary	Yes	No
Multihoming	Yes	No
Multistreaming	Yes	No
Path MTU discovery	Yes	Yes
Protection against SYN flooding attack	Yes	No
Half-closed connections	No	Yes

sctp socket api specifications

SCTP supports two styles of socket interfaces to accommodate and support all of the possible features of the protocol. This section will outline those specifications. The basic design objectives of SCTP Socket API are:

- Maintain consistency with existing socket APIs:

  To maintain consistency with other socket APIs (UDP, TCP, IPv4, etc.) so that the system call interface for SCTP has same semantic meanings as in case of TCP socket interface.

- Support a one-to-many, UDP, style interface:

  This style of interface is similar to that of UDP. The reason for this is to exploit all the possible features in SCTP such as sending data unreliably with no ordering. The outbound association setup is implicit and there is a one-to-many relationship between socket and association. Following is the typical sequence of socket calls used in the server and client.

  Server: socket(), bind(), listen(), recvmsg(), sendmsg(), close()

  Client: socket(), sendmsg(), recvmsg(), close()

It is important to note here that because of the connection-oriented nature of SCTP, multicast or broadcast communications are not supported as opposed to UDP.

- Support a one-to-one, TCP, style of interface:

  This interface supports connection-oriented, TCP, style of interface. The interface must support a single SCTP association as in TCP. One of the purposes of this interface is allow developers to port TCP applications to SCTP with very little effort. Following is a typical sequence of socket calls:

  Server: socket(), bind(), listen(), accept(), recv(), send(), close()

  Client: socket(), connect(), send(), recv(), close

The accept() call blocks until a new request is made by client to setup an association. It returns a new socket descriptor and this descriptor is used to communicate with the client using recv() and send() calls.

Since SCTP supports both UDP and TCP style sockets, some of the features cannot be mapped to existing socket interfaces. One such example is bind() system call. In one-to-many style interface, an SCTP endpoint can be associated with multiple addresses and bind() cannot be called multiple times to associate multiple addresses to an endpoint. The SCTP socket interface specification introduces a new system call sctp_bindx() to overcome this limitation. Readers are encouraged to see references to explore the semantics of the system calls in reference to SCTP mentioned in this section

SCTP IMPLEMENTATIONS

SCTP is currently in the research phase, but there are many implementations available for mainstream operating systems including FreeBSD, OpenBSD, NetBSD, Solaris, Linux, AIX and HP-UX. Moreover, user space implementations also exist for Solaris, Linux, VxWorks, Windows and proprietary platforms such as Cisco, Nokia and Siemens.

The Linux kernel SCTP (lksctp.sourceforge.net) is an open source implementation of SCTP in the Linux kernel (still in an experimental phase) under the GNU General Public License. The project was started by one of the original designers of SCTP, Randall Stewart. The current version of lksctp supports Linux kernel 2.6.14.

SCTP can be built into the kernel or can be loaded as a module. In order to test the Linux kernel SCTP reference implementation, you need to download and install lksctp tools, which should be compatible with your kernel. The lksctp-tools package provides user-level C language header files and an API for accessing SCTP for SCTP application developers and a test framework and test suite for lksctp project developers.

The latest version of lksctp tools (1.0.4) runs on Linux kernel 2.6.14. To test and run sample SCTP applications, you are first required to grab kernel 2.6.14 and compile it by enabling SCTP. After loading the new kernel, download and install the following packages from
http://lksctp.sourceforge.net/

lksctp-tools-1.0.4-1.i386.rpm

includes SCTP run-time library, sample SCTP applications

lksctp-tools-devel-1.0.4-1.i386.rpm

includes SCTP header files, SCTP man pages and source code for sample SCTP applications

lksctp-tools-doc-1.0.4-1.i386.rpm

includes SCTP RFCs and Internet drafts.

After a successful installation, you will find three sample SCTP applications: sctp_darn, sctp_test and withsctp. The sample applications sctp_darn and sctp_test can be used to test Linux kernel reference implementation of SCTP. Withsctp is a tool that can be used to replace existing TCP binaries with SCTP.

sctp applications, products and services

Most of the commercial products that are using SCTP are signaling transport solutions, since SCTP was primarily designed for transporting signaling traffic.

With its new, attractive features not supported by both TCP and UDP, SCTP is now used by many vendors for their signaling solutions. While it is still in the early phases for non-signaling applications, it has a promising future because of its prominent features and new extensions.

REFERENCES

RFCs: 2960, 3257, 3286, 3578
Internet drafts: draft-ietf-tsvwg-sctpsocket-11.txt

http://lksctp.sourceforge.net
http://www.sctp.org
http://www.sctp.de
http://www.sctp.be

RAJA HAMMAD IS THE GENERAL MANAGER OF ADVANCED DATA NETWORKING SOLUTIONS AT SPLICED NETWORKS LLC. HE IS BASED OUT OF PAKISTAN.

(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 //

Integrating Ruby on Rails and Asterisk with RAGI

THE RUBY ASTERISK GATEWAY INTERFACE PROVIDES A FRAMEWORK FOR BRIDGING

RUBY ON RAILS WITH ASTERISK THE OPEN SOURCE PBX FOR ENHANCED CALL HANDLING

BY JOHN BUSWELL

Asterisk is an Open Source PBX solution developed by Digium. Ruby on Rails is a full-stack framework for developing database backed web applications using the Model-View-Control pattern based on Ruby. Ruby is an interpreted scripting language designed for quick and easy object-oriented programming. The Ruby Asterisk Gateway Interface or RAGI is a relatively new framework for integrating Asterisk with your Rails web application.

Please keep in mind that RAGI is just a framework. While it significantly reduces the complexity and time required to build applications that interact with Asterisk, you do still have to code the application.

For more information of Asterisk and Open Source Telephony, refer to Open Source Telephony on page 32 of Issue #1 of O3 Magazine. This article will assume that you have already configured and setup your Asterisk server. We will also assume that you are running Rails on a server with the IP address of 192.168.99.20. We will be using RubyGems, the standard Ruby package manager, to obtain Ruby related software, so if you are following along, then you will need to make sure the system has Internet access.

Our developer, Joe (who has a user name of joe on 192.168.99.20), is going to install a clean Ruby environment in his home directory. First, visit *http://www.rubyonrails.com/down* and download Ruby 1.8.2 or 1.8.4. You will also need to download RubyGems.

```
mkdir -p ~/projects/
cd ~/projects/
tar zxvfp ruby-1.8.2.tar.gz
cd ruby-1.8.2
./configure –prefix=/home/joe/projects/rubydev
make && make install && make install-doc
```

Assuming all goes well for Joe, he now has a copy of Ruby in his home directory. The last make line builds and installs Ruby and builds/installs the Ruby documentation. Next, you need to untar RubyGems.

```
tar zxvf rubygems-0.8.11.tgz
```

However, before we can use RubyGems, we need to adjust the path a little. In our example, Joe is running bash.

```
PATH=$PATH:/home/joe/projects/rubydev/bin
export PATH
cd rubygems-0.8.11
ruby setup.rb
```

A few seconds later and we're installed. Next we need to use gems to get rails and ragi. To do this, simply issue the following commands:

```
gem install rails –include-dependencies
gem install ragi –include-dependencies
```

Since this is the first time gems has been run, it may take a few minutes, because it will attempt to update the Gem source index from *http://gems.rubyforge.org*.

Once gems completes its work, Ragi can be found in /home/joe/projects/rubydev/lib/ruby/gems/1.8/gems/ragi-1.0.0.

Next, we create our rails app by running rails apps/dayofweek where dayofweek is the name of our application. We're assuming Joe has an apps directory in projects/rubydev/. To start the integrated web server, we simply run ruby script/server. Now, point a browser to http://192.168.99.20:3000/, and we're up and running.

In order for Rails to work, we need a database. Lets assume Joe's box is devoid of a database, and he downloads PostgreSQL 8.1.1. To get it running, we do the following:

```
tar jxvf postgresql-8.1.1.tar.bz2
cd postgresql-8.1.1
./configure –prefix=/home/joe/projects/postgres
make && make install
cd ~/projects/postgres
mkdir -p data
bin/initdb -D /home/joe/projects/postgres/data/
bin/pg_ctl -D /home/joe/projects/postgres/data -l \
logfile start
```

Now that we have rails, ragi and postgres running, it's time to get down to business. First, we create a ragi/ directory within the lib/ directory of our application (rubydev/apps/dayofweek/lib/ragi). Then we copy all of the ruby files (*.rb) for ragi (these are stored in rubydev/lib/ruby/gems/1.8/gems/ragi-1.0.0/) to this new ragi directory.

Next, we create a handlers directory under apps/dayofweek/app/. This is where our call handlers go. Normally, in Rails a controller is used to provide the logic for your web application. In RAGI, a phone call interaction is controlled using a handler.

In config/environment.rb, you need to add some short code to startup the RAGI server on launch as a separate thread. The code opposite takes care of that.

Now a little asterisk configuration is required in extensions.conf to send call control to your RAGI application. In our example, we will route all calls to extension 353 to our dayofweek application, and we assume that we are running on 192.168.99.20.

```
exten => 353,1,Answer()

exten =>
353,2,deadagi(agi://192.168.99.20/dayofweek/dialin
)

exten => 353,3,Hangup
```

RAGI is capable of multiple call handlers within an application, and these are routed based on a URI. For our dayofweek application, all calls to extension 353 are routed to the handler “dayofweek_handler” in the handlers directory, and the method dialin will be called when the call goes through. Additional call handlers are as easy as adding them to the handler directory and configuring asterisk to route calls to them.

```
Dependencies.mechanism = :require


# Simple server that spawns a new thread for the server

class SimpleThreadServer < WEBrick::SimpleServer

  def SimpleThreadServer.start(&block)

    Thread.new do block.call

    end

  end

end


require 'ragi/call_server'


RAGI::CallServer.new(^M:ServerType => SimpleThreadServer )
```

Next, copy any necessary sound files for the application to the Asterisk server's default sound directory. Finally, we need to code up the handler. The control of the call is handled by dialin.

Please note that I have deliberately left out the code to calculate the day, and truncated the ifelseif block in announce_day due to the limited space for this article. However, the code below does demonstrate how simple it is to build call handlers within Ruby on Rails that easily interact with Asterisk. With slightly more complex code, you could easily use RAGI to implement a system to retrieve ticket status in web based bug tracking systems by phone. It should also be noted that the names within play_sound(“”) correspond to files stored in the Asterisk server's default sound directory.

```
require 'ragi/call_handler'


class DayofweekHandler < RAGI::CallHandler

  APP_NAME='dayofweek'


  def dialin

  answer
  wait(1)

  repeatDay = true

  while (repeatDay)
    greeting
    ragiday = foo_gettoday
    announce_day(ragiday)
    repeatDay = ask_repeat_day
  end

  say_goodbye
  hangup

end

def greeting

  play_sound(“today-greeting”)

end


def announce_day(ragiday)

  if (ragiday == 1)
    play_sound(“today_is_monday”) # play the day
  elseif (ragiday == 2)
    play_sound(“today_is_tuesday”)
  elseif ...
    play_sound("today_is_sunday")
  end

end

  def ask_repeat_day

  # wait about 1 second for the caller to press a key

  return (get_data(“repeat-time”, 3000, 1).length > 0)

  end


  def say_goodbye

    play_sound(“today-goodbye”)

  end

end
```

Overall, RAGI provides a fast and simple means for interacting web applications with Asterisk. Within an hour we had a working application up and running within Rails, and that's including the time to install Ruby, Rails and Postgres.

RAGI is sponsored by Snapvine, who appear to be a new Open Source Telephony startup. You can learn more about Snapvine at *http://www.snapvine.com*. According to Snapvine's website they will be at the O'Reilly Emerging Telephony conference on January 24 thru 26 2006 in San Francisco. Etel brings the best of best in cutting edge IP telephony and how that new technology is being deployed by forward-thinking pioneers. You can find out more about Etel by visiting their site at *http://conferences.oreillynet.com/etel/*.

John Buswell is the co-founder and chief technology officer of Spliced Networks LLC. He is based out of Athens, Ohio with over 12 years experience in the IT industry. He can be reached via email (jbuswell@splicednetworks.com).

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.

PostgreSQL

POSTGRESQL IS A FEATURE-RICH OBJECT-RELATIONAL DATABASE MANAGEMENT SYSTEM

POSTGRES OFFERS A POWERFUL AND FREE ALTERNATIVE TO MYSQL, SQL SERVER AND ORACLE

BY JAMES HOLLINGSHEAD AND MATHEW J. BURFORD

PostgreSQL (also know as Postgres) is a feature-rich object-relational database management system, with a large developer and user base. It is a serious choice for many commercial and non-commercial database solutions, offering a powerful alternative to other software such as MySQL and Oracle.

PostgreSQL is a descendant of the Postgres project, the name change occurred to better reflect its support for a large part of the SQL standard. The history of Postgres spans an amazing 20 years, with its first beta release being in 1987. Today Postgres supports many advanced features such as complex queries, foreign keys, triggers, views, transactional integrity, multi-version concurrency control and more. The Postgres database management system can be extended by adding new data-types, functions, operators, aggregate functions, index methods and procedural languages.

On November 8th, the PostgreSQL community released version 8.1 which saw many great improvements with over 120 new features and enhancements added. One of the major new features being support for Roles, which allows a large number of users to be managed more efficiently. Other enhancements included two-phase commit, IN/OUT parameters, shared row locking, bitmap scan and many more. Speed and performance gain in operations was also seen, especially in dual processor systems. The automatic vacuum feature was improved, which is great news for 24/7 servers. This has been an extremely successful release for Postgres.

COMPARISON

By comparison, previous releases of Postgres have been slow on performance. This is believed to be due to the differing goals that Postgres has had during its development. Those goals have often been to develop features first, and worry about speed later if necessary. Superbly, recent releases of Postgres have taken into account various speed issues, which have made its operations comparable to some of the most popular databases. Postgres claims to be faster in some areas, and slower in others.

Postgres has all of the basic features such as joins, views, referential integrity, and encrypted connections that you would expect from a commercial database and is supported on Linux, Windows, Mac OSX, BSD, and Unix unlike MS SQL Server (Windows only) and Oracle, which does not support BSD. In addition, it also allows for things such as table inheritance, which isn't supported by Oracle, and server programming, which is discussed below.

INSTALLATION

I found the installation of Postgres to be fairly straightforward. I followed the instructions provided on the online Postgres manual. This section will explain how I set up PostgreSQL v8.1 from the source code distribution. First download the PostgreSQL v8.1 source which you should find on the URL below.

http://www.postgresql.org/download/

Unzip the source, and enter the created directory

$ tar zxf postgresql-8.1.0.tar.gz

$ cd postgresql-8.1.0

Note: If you are updating from a previous version of PostgreSQL, you must backup your database data by dumping it to a file, and restoring it once the installation is completed.

For a default configuration, enter

$ ./configure

Below are some of the configure options you can use to configure your installation; more are available in the documentation.

--prefix=/path/to/install : Override the default installation path

--with-perl : include PL/Perl

--with-tcl : include PL/Tcl

--with-python : include PL/Python

--with-openssl : Require the appropriate OpenSSL setup to be installed before proceeding

PostgreSQL must be built using GNU make, other make utilities will not work as expected.

$ gmake

As a non-privileged user, you may perform regression checks on your PostgreSQL build (optional)

$ gmake check

To install the built files, you may need to be root.

$ gmake install

CONFIGURATION

This section will configure PostgreSQL to run as a low privileged user 'postgres', under group 'postgres'. It is recommended to run PostgreSQL as a low-privileged user to combat the event that an attacker takes control of the PostgreSQL server. Low privileges will ensure the attacker is limited on the amount of harm he or she can perform.

You must enter a line similar to this to your /etc/passwd file. This may need to be done as root user and you should note that you require a valid shell to perform some administration operations as this user.

postgres:x:70:70::/var/lib/postgresql:/bin/bash

Enter this line in your /etc/group file.

postgres::70:

Make a database cluster directory, this is where all information will be stored. Change the permissions to our postgres user.

$ mkdir -p /usr/local/pgsql/data

$ chown -R postgres:postgres /usr/local/pgsql/data

Switch to the user 'postgres' and initialize the database cluster. The -W option will add local security to your database, adding password authentication.

$ su postgres

$ initdb -DW /usr/local/pgsql/data

$ createdb test

It may also be necessary to copy the default configuration file and edit it as instructed by the comments.

$ cp /usr/share/postgresql/postgresql.conf.sample /var/lib/postgresql/data/postgresql.conf

If everything went alright, you should now be able to enter the database.

$ psql test

CONNECTORS

The default installation of Postgres only comes with the C and embedded C drivers. However, connectors for a large number of other languages are available for download. The list below is by no means complete, but a more comprehensive list of drivers is hosted by the Postgres maintainers at *http://gborg.postgresql.org/browse.php* under the Drivers section.

PYTHON:
PyGreSQL
http://www.druid.net/pygresql/

RUBY:
Ruby-Postgres
http://ruby.scripting.ca/postgres/

An extension library to access a PostgreSQL database from the Ruby scripting language. Also supported by Ruby on Rails configurations.

Java:
Postgres JDBC
http://jdbc.postgresql.org/

PERL:
pgperl

http://gborg.postgresql.org/project/pgperl/projdisplay.php

A native Perl interface to Postgres.

SERVER PROGRAMMING

Unlike most relational databases, Postgres stores information about data types, functions, access methods, etc in its system catalog in addition to the standard information about databases, tables, and columns. This gives PostgreSQL the ability to let users write server-side functions in languages such as Python (PL/Python), Perl (PL/Perl) and Tcl (PL/Tcl) and PL/pgSQL, a language which resembles Oracle's PL/SQL language.

More information about the specific server programming languages mentioned above may be found in the following locations:

More information about the specific server programming languages mentioned above may be found in the following locations:

Python (PL/Python) -
http://www.postgresql.org/docs/current/interactive/plpython.html

Perl (PL/Perl) -
http://www.postgresql.org/docs/current/interactive/plperl.html

Tcl (PL/Tcl) -
http://www.postgresql.org/docs/current/interactive/pltcl.html

PL/pgSQL -
http://www.postgresql.org/docs/current/interactive/plpgsql.html

With a feature set rivaling those of the leading commercial databases and support on a wide number of platforms and programming languages, Postgres is a great option for your database needs. Its ease of setup and the ability to write server-side functions for it in various languages help it pull ahead of MySQL in the open source database arena and, for those of us who like a graphic interface for making databases, tables, etc, phppgAdmin (*http://phppgadmin.sourceforge.net/*) gives you a phpMyAdmin-like graphical frontend.

There is also no reason to be worried about adopting Postgres as your database solution because you fear a lack of commercial support. The Postgres team maintains a detailed list of companies offering support for Postgres, broken up by geographic region, at *http://www.postgresql.org/support/professional_support*. Here you can find contact information, the company's specialties, number of employees, business hours and even the languages they speak.

JAMES HOLLINGSHEAD IS THE EXECUTIVE EDITOR OF O3 MAGAZINE. HE CAN BE REACHED VIA EMAIL AT JAMES@O3MAGAZINE.COM.

LINUX ASIA 2006

Asia's Premier Open Source Conference & Expo

FEBRUARY 8-10, 2006 ■ NEW DELHI, INDIA

The Customer Speaks

- Industry Sessions
- E-Governance Forum
- Education Forum
- Technology Forum
- Technology Workshops
- LFY Awards
- Penguin Party
- .ORG Mela
- Industry Expo
- .GOV Pavilion

REGISTER TODAY

www.linuxasia.net

Beyond Intrusion Detection

GETTING AN INTRUSION DETECTION SYSTEM INSTALLED IS ONLY THE FIRST STEP; MANAGING AND MAINTAINING THE INTRUSION DETECTION SYSTEM CAN BE SIMPLIFIED THROUGH THE USE OF VARIOUS THIRD PARTY APPLICATIONS

BY JOHN BUSWELL

Last month, we took a broad look at Open Source Intrusion Detection System (IDS) solutions with a focus on Snort, the industry standard IDS solution. This month we will look at how to handle the data from the IDS, Dynamic Event handling, Front ends, how to keep your IDS rules up to date and testing your IDS. Like last month, this article is intended to provide you with a quick top down assessment of your available options. Later in the series we will focus in detail in building and configuring a complete IDS solution.

EVENT HANDLING

Barnyard works in a manner similar to Snort - it waits to receive an event from Snort and then passes the event through one or more plug-ins. When used with Barnyard, Snort is free to continue network processing since Barnyard has taken over the handling of the event. The main advantage to using Barnyard is on high-speed networks where Snort has to deal with large volumes of data. Barnyard also requires less privileges than Snort, which requires some degree of root access. Barnyard works hand in hand with some form of database and supports both Postgres and MySQL.

Barnyard supports two modes - batch and continuous processing. With batch processing, Barnyard will process a number of files and the stop. While continuous processing, Barnyard will process the files then wait for the next Snort event. This makes it possible to run Snort on one system, and utilize multiple systems running Barnyard to process the data which might be shared via NFS or another network file system.

Fast Logging Project for Snort (FloP) provides similar event handling but with a unique delivery system. FloP decouples the output plugins from snort, gathers all alerts and passes them on to a central server. This central server then collects and stores the data in a database for further processing. While the results are similar to Barnyard, the approach is innovative and offers a much faster solution in a high-speed network where multiple snort sensors are deployed. FloP can be downloaded from *http://www.geschke-online.de/FLoP/*.

dynamic event handling

While reporting a critical event so an administrator or third party application can do something about it is important, real-time dynamic event handling is far more responsive and lays the foundation for a good Intrusion Prevention System. Starting with Snort 2.3.0 RC1, an integrated IPS solution called snort_inline was added. This new capability utilizes data from iptables (Linux firewall) instead of libpcap and then generates new rules to help iptables process packets based on Snort rules. There are three rules that snort_inline outputs: drop (and log), reject (and log) and sdrop (drop and don't log). The end result is that snort can update the firewall rules in real-time based events that occur.

There are a number of other projects that provide similar functionality. Fwsnort (*http://www.cipherdyne.org/projects/fwsnort/*) offers similar iptables integration, snort2pf (*http://unixgu.ru/?go=snort2pf*) offers the same functionality but for OpenBSD packet filter, and an improved fork called snort2c (*http://snort2c.sourceforge.net/*) also exists.

FRONT ENDS

There are a number of front ends available for snort that simplify processing of snort output and/or snort configuration. Basic Analysis and Security Engine (BASE), which is available from *http://secureideas.sourceforge.net/*, is a project based off of ACID that provides a web front-end to query and performs analysis on alerts from snort. Another project, Placid (*http://speakeasy.wpi.edu/placid/*), provides similar functionality. One application, called Snort Report (*http://www.symmetrixtech.com/download.html*),

provides a quick and straightforward solution if you are looking for simple reporting capabilities.

SnortSMS provides an excellent web based frontend for remotely administering Snort and Barnyard based IDS solutions. It can push configuration files, manage rules and monitor the system's health and statistics and is an excellent solution for unifying multiple snort sensors. SnortSMS can be found at *http://snortsms.sourceforge.net/*.

SGUIL (*http://sguil.sourceforge.net/*) provides a GUI based Analyst console for network security monitoring. It works in a client/server configuration with a single SQUIL server which interacts with the snort sensors and many gui clients which interact with the SQUIL server. SQUIL's primary advantages over other ACID based solutions are speed and the capability to do advanced queries. Unlike the web based frontends, SGUIL is a GUI application.

rules and updates

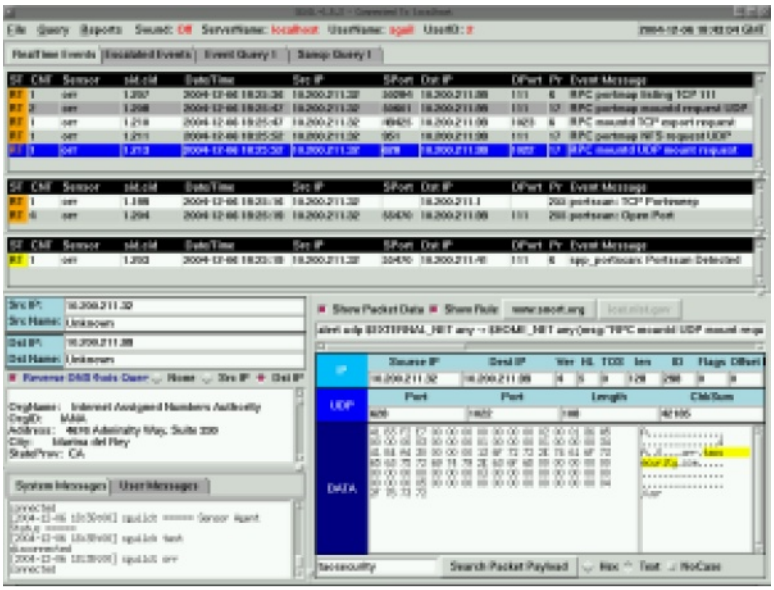
As we mentioned last month, the rules are the heart of your IDS solution. Keeping those rules up to date is a critical task. Oinkmaster (*http://oinkmaster.sourceforge.net*) is a perl script that will update and manage snort rules. It is relatively simple to configure - just edit oinkmaster.conf to define the necessary settings and sources for rules, and then put oinkmaster.pl -o /etc/snort/rules into the crontab. Also worth a mention is SigTranslator, a project designed to convert IDS signatures between different formats. It can be found at *http://translator.b59.net/.*

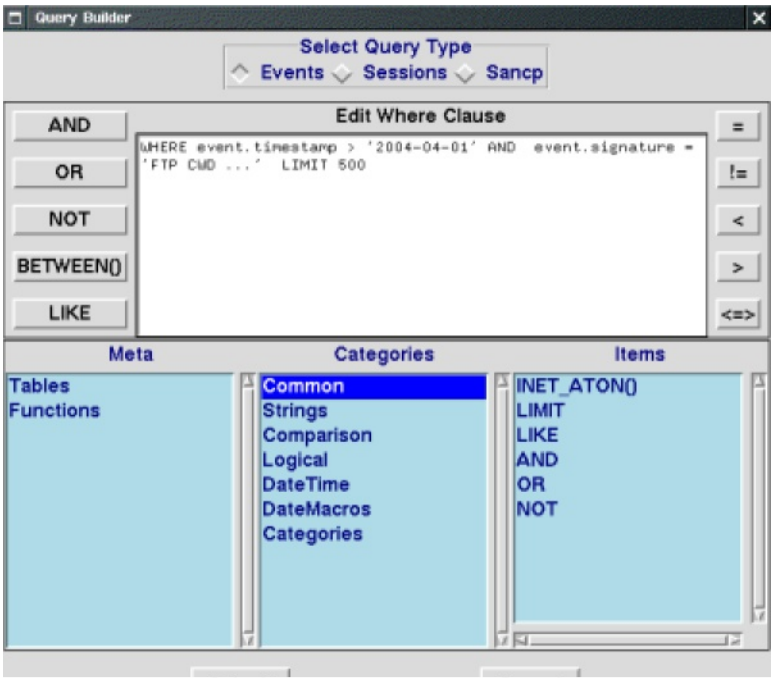
TESTING THE IDS

Once you have your IDS in place, get the event handling setup and are performing analysis on the data from the IDS, you will most likely want to test your IDS rather than wait for an attack to do it for you. The best way to test your IDS is to use any of the widely used vulnerability assessment tools. Such tools include nmap (*http://www.insecure.org/nmap*), yersinia (*http://yersinia.sourceforge.net/*), Nessus (*http://www.nessus.org*), and Dsniff (*http://www.monkey.org/~dugsong/dsniff/*) , or you could craft your own tests tools such as ScaPY (*http://www.secdev.org/projects/scapy/*). Any of these solutions will work very well for testing purposes.

NEXT

In the next issue, we will look at load balancing snort in a IDS load balancing solution.





SGUIL SCREENSHOTS - EVENTS (TOP), QUERY BUILDER (BOTTOM)

>THIS IS **THE WAY**

600 MILLION PEOPLE MOVE AROUND THE PLANET.

You'll find Nortel™ in every single one of the world's top twenty airlines. And wherever secure, reliable data and voice communications are most critical.

>THIS IS NORTEL™

www.nortel.com/commerce