INSIDE: System Management with Levanta's Intrepid-M

# o3:

The Open Source Enterprise Data Networking Magazine

Issue 4
Released Q1 2006

http://www.o3magazine.com

# Prioritizing Network Traffic
# A business look at controlling convergence with Open Source


**Prioritizing Voice and Multimedia with Linux QoS**

**Deploying a Secure Internet Presence**

**Dynamic Routing Protocols**

**Intrusion Detection with SNORT**

**Secure DNS Deployment for system administrators**

# doing business in Europe ?

# are your servers close enough to your target market ?

# high performance Internet Services

## colocation
## dedicated servers
## web hosting


**Layer Two**
http://www.layer-two.net

+44 870 141 7273
sales@layer-two.net

## CONTENTS

**Issue 4**
**Released Q1 2006**
**http://www.o3magazine.com**

### @o3 MAGAZINE



### SECURITY



Greg Jordan looks at deploying secure internet services from a business perspective. Article examines both physical and logical security.

### INTERNET TECHNOLOGIES



Greg Jordan introduces dynamic routing protocol concepts, compares policy routing, QoS and other techniques. Part 1 of a series.

### BUSINESS SOLUTIONS



Need to justify QoS and packet classification solutions to management, or simply trying to understand why QoS is needed, this is for you!

### WEB TECHNOLOGIES



Bharat Shetty explains RRDtool. This article introduces the popular graphing tool and looks at lighttpd's integrated rrdtool features.

### VOIP / VIDEO COMMUNICATIONS



Muhammad Hammad looks at QoS and its role in helping prioritize VoIP communications across IP data networks. This article focuses on Linux 2.6 based QoS and the optimum configurations for VoIP prioritization.

### IP NETWORKING



John Buswell looks at Bind 9.3.2 and walks us through a secure installation of the popular DNS server. Configuration and setup included!

### NETWORK APPLICATIONS



David Dennis of Levanta looks at the pain of Linux Systems Management and offers the Levanta Intrepid-M as one possible solution.

### NETWORK SECURITY



Naveen Sharma walks us through the design, concepts, architecture and installation of Snort, the Open Source IDS solution.

### MOBILE TECHNOLOGIES

Issue #5 of o3 magazine debuts our new mobile technologies column.

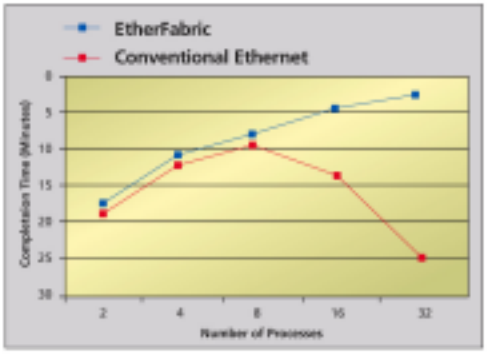
Visit our new site on or after March 22nd and check out a bonus Opera Mini interview!!

# ACCELERATE APPLICATION PERFORMANCE!


**>> HALF THE LATENCY**

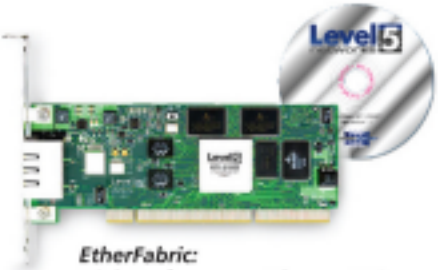
**>> TWICE THE BANDWIDTH**

**>> 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


CANADA ON RAILS

# YVR06

DAVID HANSSON
THOMAS FUCHS
DAVE ASTELS
DAVID BLACK
JOE O'BRIEN
JAMES ADAM
STEVEN BAKER
MICHAEL BUFFINGTON
ROBBY RUSSELL
GEOFFREY GROSENBACH
KYLE SHANK
JEREMY VOORHIS
ALEX BUNARDZIC
SEBASTIAN KANTHAK
AMY HOY

## VANCOUVER, BRITISH COLUMBIA

ARPIL 13-14, 2006

TICKETS ON SALE NOW
www.canadaonrails.com

# A New Look For o3 Magazine

**ARMED WITH A NEW RELEASE OF SCRIBUS, THREE MONTHS OF FEEDBACK AND EXPERIENCE WE HAVE UNDERTAKEN A MAJOR OVERHAUL OF O3 AND WE HOPE YOU LIKE THE RESULTS...**

By John Buswell

Welcome to Issue 4 of o3 magazine. As you have probably noticed, this issue looks considerably different and in my opinion, considerably better than the first three issues.

Many thanks to the folks over at the Scribus project for a cool 1.3.2 release, which has helped improve o3 magazine. If you're not already aware, o3 magazine is produced using open source software exclusively. Unlike some commercial publications that cover open source but use non-open source projects, we're 100% behind the technologies we are bringing to you each month.

I would like to bring to your attention the Canada on Rails conference, which is our featured event this month. The Canada on Rails event is the first Ruby on Rails focused event of its kind. We wish them all the best, and if you can make it, they have an excellent line up of speakers and workshops.

On behalf of our entire team, I would like to apologize for the delay in getting this issue out there. I made the decision to delay this issue, so that we could review the publication, make changes, process the feedback from the thousands of readers that submitted their opinion on the magazine.

We have a new editorial team, I would like to welcome Michelle Jordan to the o3 team. Unfortunately we had to bid farewell to James Hollingshead, James has worked hard to keep me in check over the past couple of months and we wish him all the best in his future ventures.

Finally, I would like to wrap up this month's editorial by announcing our new web site which goes on-line on mid-March 2006. Issue 5 and Issue 6 will be hot on the heels of this issue!!!

**o3 magazine**

John Buswell
***Publisher and Editor in Chief***

Greg Jordan
***Managing Editor***

Michelle Jordan
***Contributing Editor***

Gaston Thauvin
***Cover Photograph***

Stew Benedict
Shawn Wilson
Raja Hammad
Frank Boyd
John Buswell
Bharet Shetty
***Technical Review Panel***

***Advertising Information:***
Andrew Costello
acostello@o3magazine.com

***Publisher Information:***
o3 magazine is published monthly by Spliced Networks LLC and distributed free of charge to the public.

**eclipsecon 2006**
March 20th - 23rd 2006
Santa Clara, California, USA
http://www.eclipsecon.org/

**spring <br /> 2006**
March 23rd 2006
Athens, Ohio, USA
http://www.seomug.org/conference.cfm

**PHP Quebec 2006**
March 29th - 31st 2006
Quebec, Canada
http://conf.phpquebec.com/en/conf2006/

**LinuxWorld Conference & Expo**
April 3rd - 6th 2006
Boston, MA, USA
http://www.linuxworldexpo.com/

**7th International Free Software Forum**
April 19th - 22nd 2006
Porto Alegre, Brazil
http://fisl.softwarelivre.org

**Penguicon 4.0**
April 21st - 23rd 2006
Livonia, Michigan, USA
http://www.penguicon.org

**Linux on Wall Street Show and Conference**
April 24th 2006
New York, New York, USA
http://www.linuxonwallstreet.com

**MySQL Users Conference**
April 24th - 27th 2006
Santa Clara, California, USA
http://www.mysqluc.com

**ApacheCon Europe 2006**
June 26th - 30th 2006
Dublin, Ireland
http://www.eu.apachecon.com/

**FEATURED EVENT**


Canada on Rails will be held on April 13th - April 14th in Vancouver, Canada. This event will be the first Ruby on Rails conference.

The event showcases 15 speakers from around the world including David Hansson, creator of Ruby on Rails.

On Day 1, the event opens at 07:30am, with a keynote from David Hansson opening the event at 09:15am. This is followed by Enterprise SOA with Rails by Joe O'Brien at 11:00am.

The afternoon covers topics such as IDE/Tools, AJAX, the benefits of developing Rails applications test-first, and a presentation on the reduction of code necessary to supply typical solutions to business problems with Rails.

The second day continues the high quality talks and presentations from Day 1. The event starts at 08:00am on Day 2 with Advanced Rails AJAX Techniques given by speaker Thomas Fuchs.

The second day continues with Engines: Team Development with Rails, and Generating Great Graphs with Ruby on Rails.

The afternoon rounds up the even with a look at Internationalization, FileColumn, and Using Ruby on Rails to Make a Massive Multiplayer Game.

Sneaking Rails Into The (Legacy) System, on Day 2 of the event will be of interest to any developer considering the use of Rails as a replacement for an existing database driven application.

**http://www.canadaonrails.com**


## trac 0.9.4
*http://www.edgewall.com/trac/*

This issue's featured open source software project is trac. Trac is an enhanced wiki and issue tracking system for software development projects.

Trac uses a minimalistic approach to web-based software project management.

Trac provides an integrated system for managing software projects, an enhanced wiki, flexible web-based issue tracker and an interface to the subversion revision control system.

One of the most important features in Trac is its capability to easily interface with Subversion. Subversion is a next generation revision control system for software projects, and is often being used as a replacement for CVS.

Trac is an excellent project for small and medium sized companies that want a stable, and well designed system for managing software projects.

Trac is designed to let software developers work efficently by minimizing the effort associated with managing software projects.

## Typo
*http://www.typosphere.org*

Typo is a lean engine that makes blogging easy. Typo has a web based admin interface that handles configuration and management. Typo runs under Ruby on Rails.

## RTRails
*http://rubyforge.org/projects/rtails/*

RTRails is Realtime on Rails, a web based collaborative environment. It is designed for small or mid-sized groups for communication, planning and includes a chat system. It is an AJAX based solution.

## RailsTidy
*http://www.cosinux.org/blogs/dam/pages/rails-tidy*

RailsTidy is a plugin for Ruby on Rails that allows validation of rhtml templates, html output of functional tests and to clean the html generated by rails. It takes advantage of both Tidy and Ruby Tidy open source projects.

# 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**

# Deploying a Secure Internet Presence

**IN TODAY'S BUSINESS WORLD DEPLOYING INTERNET SOLUTIONS IS A NECESSARY MEASURE TO ATTRACT NEW CUSTOMERS AND PROVIDE SERVICES TO EXISTING CUSTOMERS DEPLOYING THOSE SOLUTIONS SECURELY IS A CRITICAL TASK FOR ANY BUSINESS...**

By Greg Jordan

For most modern businesses, the Internet has become a crucial part of daily business operations. The Internet is used to generate new business leads, support existing customers, or even to deliver products and services. Deploying a secure Internet presence and maintaining that security is a critical factor in remaining competitive in todays business world.

In the previous issue of o3 magazine, we discussed how to deploy a global Internet presence, and the necessary decisions and the reasoning behind using services such as co-location to push localized content closer to the target markets. This month we look at issues that should be addressed to your satisfaction when selecting a co-location facility, though the issues raised also apply to your local IT facilities.

This article takes a layered approach to addressing the security concerns related to maintaining off-site and on-site Internet services. Many businesses will have some of the measures suggested in place already. It is important to take into account that lawmakers, not only in the United States but throughout the world, are putting into place legislation to protect their citizens' data whenever it is stored or transmitted electronically. You may not even reside in the state or country involved, but if your customers reside there, you may be subject to their laws whether you like it or not.

A good example, right here in Ohio is state law that went into effect earlier this month that requires the proper security and handling of Ohio residents personal information, if that information could be used to cause those residents direct losses (such as financially through identity theft). Lets take a small company that operates an on-line store, that on-line store might be hosted on a server in a data center at some web hosting company. While the company is not directly responsible for the management of the server, they are directly responsible for the security and protection of their customers' data. While it is good business practice to protect this data, lawmakers have decided to back up that type of common sense with strict fines and penalties. The Ohio law takes a few steps further, requiring the company to publicly notify its customers of the breach of data. There is nothing more your competitors would like than you having to pay for an advertisement that admits you have poor security systems in place.

If you are an IT professional, perhaps a web developer, consultant or system administrator, there is now a strong possibility (especially if you are an independent contractor) that if you simply roll out a solution for a client that involves the management of their customer data, that you be indirectly liable for the security of that data. Perhaps you weren't involved in the hosting of that site, but if it was your software that was used to store the data, the client may be looking for someone to blame. Did they pay for your services to deploy a secure solution? Did you advise the customer in writing that they needed to have the software continuously maintained and hosted on a system that was actively administered by a trained professional? One data breach could result in your client's image being ruined- and perhaps even in the destruction of their business. You could quickly find that your lack of hosting guidelines- or even the software itself- becoming a legal target for

that business trying to recoup their costs, or trying to direct blame away from their company.

A more precarious situation might be if you used a hosting provider where you obtain a regular commission from their hosting. Do you regularly check to make sure your clients are being taken care of? With far too many full-time technical support agents turned part-time entrepreneur, do you really know who is hosting your Internet services?

It doesn't even need to be a mere part-time operation to raise concern. Some highly professional businesses- such as those run by seasoned business people who unfortunately place too much faith in high price, brand name solutions, and who favor rapid deployment over application security- can be equally insecure.

### PHYSICAL SECURITY

Before you start to look at packets, firewalls, and application security, physical security should be the first line of defense. There is no point implementing a state-of-the-art security system if someone can photocopy an ID badge, walk right into your data center and walk out with your data on your own disks!! Whether it is your own data center or that of a co-location company, getting details on physical security is a good thing.

Is the building in a secure location? If the building is in a bad part of town or an area with a high crime rate, it may not be the best location. Typically, the area in which the building is located should give you some idea as to the degree of physical security measures that should be in place. Is there a security guard on duty 24/7/365? Is key card access required to enter the building after hours, on holidays? Is a special key-card access required to enter the floor where the servers are located? Is the building's power system secure? There is no point in having state of the art security if the fuse box in the basement is left unsecured.

If you're at a shared hosting facility, are visiting customers escorted while on the premises? Do customers have access only to their equipment or the entire floor? Are UPS solutions shared or dedicated? Do employees have to follow a checklist when maintenance on adjacent racks is performed?

A simple matter of an employee unplugging the wrong customer from a UPS, or unhooking the wrong port from a shared switch can result in critical down time. In fact, a check to make sure 802.1x port-based access control is implemented on the network, to prevent legitimate users gaining access to the network with unauthorized devices, is a good idea as well.

Physical security, however, doesn't stop at consideration of the risk of malicious users or theft. Other issues, such as fire, also put your Internet services at risk. As such, making sure sufficient fire suppression systems are in place and regularly maintained is important.

These checks should not just be performed when you sign up. Each time an employee visits the co-location site, or when your account executive calls seeking additional sales, check with them to see if the security measures are all still in place, if any improvements have been made, or if any security breaches or problems have occurred recently.

### PHYSICAL NETWORK SECURITY

Physical network security starts at Layer 1, the cabling. Something as simple as a decent cable management solution will add a degree of security to the cabling system. This could be as simple as tie-wrapping flexible conduit once the cabling has been installed, or running cabling in conduit that is high enough to require equipment in order to reach. These steps make it a lot harder for a malicious employee or other users to cut a cable in order to cause a serious network outage.

Port-based access control uses the hardware MAC address of the client device, along with potentially other measures to authenticate a users access to the network. Solutions such as dynamic PVID / Port VLAN membership prevent a user from attempting to move their equipment from one port to another in order to gain higher degrees of network privileges. Strict port-based access control can limit new devices from even accessing the network without prior permission from the network administrator. Such features are important, again to prevent physical security violations.

### LOGICAL NETWORK SECURITY

In a switched environment VLANs are used to restrict the flow of data across a shared resource, such as a switch. In a shared hosting environment, such as co-location or dedicated hosting, it is desirable to have your equipment connected on its own VLAN, perhaps with its own IP subnet. This can prevent another customer either accidentally or deliberately assigning their equipment the same IP addresses assigned to your equipment.

Stateful firewalls running on servers directly, as well as a network firewall strategy, can provide a

good first line of defense. If you're running a web server remotely, it is likely that you will need to admin the system remotely. Making sure that you use an up-to-date release of OpenSSH, strong encryption algorithms, and limit SSH access to only a small number of necessary subnets via the firewall will greatly enhance the security of your server. Placing the firewall rules on the server provides a degree of redundancy in the event that human errors result in firewall rules being relaxed that could put your services at risk. Simply running a command such as netstat -nap will give you an idea of the open ports on the system, whether you need them or not, etc.

#### SITE REDUNDANCY

Maintaining a secure Internet presence means that you need to make sure your services are up 24/7/365, and that at least one authoritative server is in control at any one given moment. If it's possible for your services to go down due to a single attack against one network, or because of a physical disaster, such as a flood or even a major power outage, then you need to deploy some form of site redundancy. This could be a simple as having a single server co-located outside of your network. DNS is a critical part of site redundancy, and is discussed further in this month's DNS article.

#### SMTP

Whether you use email to contact your customers, to run distribution lists, or simply utilize it as a means of communication between employees, email is vulnerable to a wide range of outside problems, ranging from SPAM and Viruses to Phishing attempts. A good SMTP system will utilize a primary mail deliver site, with one or more remote sites that perform proxying.

There are a wide range of tools for integrating anti-virus scanning into email solutions. ClamAV is a widely used open source solution. Combined with solutions such as DSPAM, whitelist, greylist and blacklist techniques, it is possible to eliminate a high number of threats. Combine this with a good corporate policy on email usage, and it can result in a simple and secure SMTP solution.

#### HTTP / HTTPS

Deployment of a dedicated web server rather than a shared system is a good idea, as it reduces the number of ways a server can be compromised from the outside. Even on a dedicated web server, techniques such as chrooting apache, which has been made easy with solutions such as modsecurity, is a must. Limiting dynamic scripts, and using static content where possible- while maintaining a close eye on what is happening on your web server through log analysis is a must.

In most solutions, some customer data will be stored locally and/or transmitted to some form of payment processing gateway. It is vital that you only store the information that is necessary for the transaction, and once that transaction has been approved, that only the necessary elements for tracking the transaction are kept. If that information needs to be transmitted, it should be done over a secure medium such as SSL.

#### ENCRYPTED FILE SYSTEMS

When storing customer information locally, it should be stored in the database in an encrypted manner, not as plain text. Ideally, that database should be stored on some form of encrypted file system, so that in the event that the disk is physically stolen, the customer data is still not possible to reconstruct.

An extra level would be to encrypt the local swap partitions, to prevent attacks which might push vital system data into swap where it could then be possibly read or manipulated to compromise the system.

#### CONCLUSION

While the traditional focus of deployment of secure Internet resources will focus on firewalls, intrusion detection, packet inspection, and intrusion prevention technologies, it is very important not to lose sight of common sense issues such as physical security. As contract IT professionals, it is equally as important to make sure the customer is aware of the responsibilities of maintaining a secure Internet presence, so that even long after you have left the scene, you cannot be held liable for not informing the customer of their duties.

***Greg Jordan is Managing Editor at o3 magazine. Greg has over two decades experience in the Information Technology industry with a focus on Internet Service Providers and Telecommunication Carriers.***

# >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.

# spring <br /> conference 2006

SEOMUG SOUTHEAST OHIO MACROMEDIA USER GROUP

ITAAO

OHIO UNIVERSITY

Adobe

## Thursday, March 23, 2006

The Midwest's Premier One Day Designers & Developers Event!

*"Spring Break is one of the better user group regional events in existence. The speakers and sessions are second to none and the conference price of $25 is unmatched in the industry."*

Ed Sullivan, Program Manager, Developer Relations Adobe Systems

**Produced by the Southeast Ohio Macromedia User Group**
**In partnership with the IT Alliance of Appalachian Ohio, Ohio University and Adobe Systems**

**Admission is $25, which includes a Lunchtime session by Tim Buntel in which we provide your lunch!**

**Online Registration Beginning Feb 1st or Tickets At The Door** **Website: www.seomug.org**

**Presenters Include:** (name / topic)

- **Phillip Kerman : Flash**
- **Rob Gonda : AJAX**
- **Carolyn Snyder : User Experience**
- **Dan Dura : ActionScript**
- **Glenda Vigoreaux : Dreamweaver (Hands On Sessions)**
- **John Cummings : ColdFusion**
- **Neil Ross : Development Patterns**
- **Joe Lowery : Dreamweaver**
- **Gary Kraeger : Enterprise Email**

**Many More**

**Keynote by Tim Buntel**
**Product Manager**
**Adobe Systems**

**Check the Website beginning Feb 1**
**www.seomug.org**

**Tracks Include:**

- **Design & User Experience**
- **Programming & Development**
- **3D Gaming & Simulations**
- **Adobe & Related Products**

**Conference bookstore with author signings**
**SEOUG Community Suite**
**Wireless Access Provided**
**Email Cafe**

**Prizes - Prizes - Prizes - Prizes**

Watch the website:
Prizes for early bird registration:

**www.seomug.org**

# Dynamic Routing Protocols

**DYNAMIC ROUTING PROTOCOLS ENABLE A NETWORK TO RESPOND TO CHANGES BOTH INSIDE AND OUTSIDE OF THE NETWORK. THEY PROVIDE THE CAPABILITY TO BUILD HIGHLY AVAILABLE NETWORKS THAT CAN RESPOND TO BOTH LINK FAILURE AND CHANGES IN NETWORK TRAFFIC**

By Greg Jordan and Wikipedia.com

Dynamic Routing Protocols are used to automatically update the preferred path that packets take over a network, based on a router's response to conditions on the network. There are a number of dynamic routing protocols, the most widely-used on the Internet is BGP (Border Gateway Protocol). These dynamic routing protocols are important because they help to build fault tolerant and high performance networks that are necessary in today's business world. This article is intended to provide a very basic, high-level view of dynamic routing and IP routing in general.

### BASICS

Some basic information regarding IP networking is required. The following information on Subnetworks, Network Masks and Subnetworking has been provided courtesy of Wikipedia, the free encyclopedia. Due to the inclusion of this information, this article is being released under the terms of the GNU Free Document License.

### SUBNETWORKS

The word subnetwork (subnet for short) has two related meanings. In the older and more general meaning, it meant one physical network of an internetwork. In the Internet Protocol (IP), a subnetwork (usually known as a subnet) is a division of a classful network. The rest of this article is about the second meaning.

Subnetting an IP network allows you to break down what appears (logically) to be a single large network into smaller ones. It was originally introduced before the introduction of classful network numbers in IPv4, to allow a single site to have a number of local area networks. Even after the introduction of classful network numbers, it continued to be useful, as it reduced the number of entries in the Internet-wide routing table (by hiding information about all the individual subnets inside a site). As a side benefit, it also resulted in reduced network overhead by dividing the parts which receive IP broadcasts.

### NETWORK MASKS

A network mask, also known as a subnet mask, netmask or address mask, is a bitmask used to tell how how many bits in an Octet(s) identify the subnetwork, and how many bits provide room for host addresses.

Subnet masks are usually represented in the same representation used for addresses themselves; in IPv4, dotted decimal notation - four numbers from zero to 255 separated by periods, e.g. 255.128.0.0.

But in subnet masks only some of the numbers are allowed: 0,128,192,224,240,248,252,254,255

Less commonly, it can be represented as an eight-digit hexadecimal number (e.g. FF.80.00.00 = 255.128.0.0).

A shorter form, which is known as Classless Inter-Domain Routing (CIDR) notation, gives the network number followed by a slash and the number of 'one' bits in the binary notation of the netmask (i.e. the number of relevant bits in the network number). For example, 192.0.2.96/28 indicates an IP address where the first 28 bits are used as the network address (same as 255.255.255.240).

### SUBNETTING

Subnetting is the process of allocating bits from the host portion as a network portion. For example, giving the class A network 10.0.0.0 a subnet mask of 255.255.0.0 would break it down into 256 sub-networks (10.0.0.0 to 10.255.0.0). Indicating that the first octet of the IP address shows the network address, the second one shows the subnet number and the last two show the host part. A bitwise AND operation of the host address with the subnet mask extracts the complete subnetwork address (see example below).

Subnet masks are not limited to whole octets, either. For example 255.254.0.0 (or /15) is also a valid mask. Applied to a class A address this would create 128 subnetworks in intervals of two

### WHAT IS ROUTING?

Routing is essentially the forwarding of packets from one network to another. The Internet, and most corporate networks, are made up of many smaller subnetworks called subnets. An IP subnet is a subset of a larger network, and is defined by the network mask. The network mask defines the range or size of the subnet. For every subnet, there is a network address (at the start of the subnet) and a broadcast address (the end of the subnet). The broadcast address is used on Ethernet networks for example to send a special broadcast packet which all devices on the network are expected to respond to.

Typically, a router will have an interface (logical or virtual) in the subnet. The clients on that subnet are configured to pass packets to the router. This is usually done by sending the IP packet destined for a remote address but with the MAC address of the router's interface. The packet is then picked up by the router, which uses the destination IP address to determine how it should route the packet, if it can at all.

In its basic form, a router could have an address in two subnets, lets say 192.168.1.0/24 and 10.1.2.0/24 are the two subnets. The router may exist as 192.168.1.1 and 10.1.2.1 in each subnet. A client 192.168.1.10 wants to send a packet to 10.1.2.100. To do this, it will send the packet destined to 10.1.2.100 but with the MAC address of the 192.168.1.1 interface. When the router receives the packet, it looks in its routing table to see how to forward the packet. If it finds a suitable route, it will forward the packet to the destination by changing the destination MAC address to that of 10.1.2.100 and forward the packet out the interface associated with 10.1.2.1.

If the router doesn't have a suitable route, it will send a special ICMP response back to the source IP (192.168.1.10) advising that the destination host or network is unreachable.

### ROUTING TABLE

The routing table in some form or another exists on all IP capable devices. At the very least, the routing table provides destination and gateway information. The gateway being the IP to forward packets to that destination. Sometimes the routing table may contain specific information such as the interface to send the packets out. It may also contain what is called a metric. A metric is a value which gives a particular route priority. Metrics are important because they enable you to prefer one route over another, but if the higher priority route becomes unavailable, the backup route is still in the table to prevent routing failures.

### POLICY ROUTING

Policy routing is a term you probably have come across. To avoid confusion between policy routing and dynamic routing protocols, a brief primer on policy routing is provided. Policy routing involves tagging packets that have been classified with a specific priority or type of service, or to match attributes about a specific packet and make routing decisions based upon those attributes.

Dynamic routing protocols affect overall routes on the network based on network conditions or configuration, rather than specific packet attributes. For example, with BGP routes that are announced, BGP will change the routing table to prefer one route over another based on the conditions of the network exchanged between two BGP peers. Policy routing may change how a specific packet or stream of packets is routed over the network based on the source port or original of the packet.

### AUTONOMOUS SYTEMS (AS)

In the Internet, an autonomous system (AS) is a collection of IP networks and routers under the control of one entity (or sometimes more) that presents a common routing policy to the Internet. See RFC 1930 for additional detail on this updated definition.

Originally, the definition required control by a single entity, typically an Internet service provider or a very large organization with independent connections to multiple networks, that adhere to a single and clearly defined routing policy. See RFC 1771, the original definition (now obsolete) of the Border Gateway Protocol. The newer definition of RFC 1930 came into use because multiple organizations can run BGP using private AS numbers to an ISP that connects all those organizations to the Internet. Even though there are multiple autonomous systems supported by the ISP, the Internet only sees the routing policy of the ISP. That ISP must have a public, registered ASN.

A unique AS number (or ASN) is allocated to each AS for use in BGP routing. With BGP, AS numbers are important because the ASN uniquely identifies each network on the Internet.

### ROUTING PROTOCOLS

There are a number of routing protocols – IGRP, EIGRP, RIP, OSPF, IS-IS and BGP. Routing protocols are typically interior or exterior to

an AS. Interior routing protocols are used to exchange information and maintain routes within a single AS. Exterior protocols, such as BGP, are used to exchange data between multiple AS. IGRP and EIGRP are Cisco solutions. RIP is available in version 1 and version 2. RIP is a limited and basic protocol, ideal for smaller networks who do not wish to add the complexity of OSPF to their network.

#### OSPF

Open Shortest Path First (OSPF) is a link-state, hierarchical Interior Gateway Protocol (IGP) routing protocol. Dijkstra's algorithm is used to calculate the shortest path tree. It uses cost as its routing metric. A link state database is constructed of the network topology which is identical on all routers in the area.

OSPF is perhaps the most widely used IGP in large networks. It can operate securely, using MD5 to authenticate peers before forming adjacencies, and before accepting link-state advertisements. A natural successor to RIP, it was VLSM capable or classless from its inception. A newer version of OSPF (OSPFv3) now supports IPv6 as well. Multicast extensions to OSPF (MOSPF) have been defined, however these are not widely used. OSPF can "tag" routes, and propagate these tags along with the routes.

An OSPF network can be broken up into smaller networks. A special area called the backbone area forms the core of the network, and other areas are connected to it. Inter-area routing goes via the backbone. All areas must connect to the backbone; if no direct connection is possible, a virtual link may be established.

Routers in the same broadcast domain or at each end of a point to point link form adjacencies when they have discovered each other. The routers elect a designated router (DR) and backup designated router (BDR) which act as hub to reduce traffic between routers. OSPF uses both unicast and multicast to send 'hello packets' and link state updates. Multicast addresses 224.0.0.5 and 224.0.0.6 are used. In contrast to RIP or BGP, OSPF does not use TCP or UDP but uses IP directly, using IP protocol 89.

#### BGP

The Border Gateway Protocol (BGP) is the core routing protocol of the Internet. It works by maintaining a table of IP networks or 'prefixes' which designate network reachability between autonomous systems (AS). It is described as a path vector protocol. BGP does not use technical metrics, but makes routing decisions based on network policies or rules. The current version of BGP, BGP version 4, is specified in request for comment RFC 4271 (as per Jan 2006). This RFC obsoletes RFC 1771.

BGP supports classless interdomain routing and uses route aggregation to decrease the size of routing tables. Since 1994, version four of the protocol has been in use on the Internet; all previous versions are considered obsolete.

BGP was created to replace the EGP routing protocol to allow fully decentralized routing in order to allow the removal of the NSFNET Internet backbone network. This allowed the Internet to become a truly decentralized system.

Very large private IP networks can also make use of BGP; an example would be the joining of a number of large Open Shortest Path First (OSPF) networks where OSPF by itself would not scale to size. Another reason to use BGP would be multihoming a network for better redundancy.

Most Internet users do not use BGP directly. However, since most Internet service providers must use BGP to establish routing between one another, it is one of the most important protocols of the Internet. Compare and contrast this with Signalling System 7, which is the inter-provider core call setup protocol on the PSTN.

#### OPEN SOURCE IMPLEMENTATIONS

There are a number of open source implementations including OpenBGPD, Quagga, Xorp, GNU Zebra and BIRD.

#### FURTHER ARTICLES

This article was designed to introduce the concepts behind dynamic routing protocols to you. In issue 7 of o3 magazine, the network column will continue on from this article looking at the Vyatta router platform based off Xorp and another article in the same issue will look at configuring zebra to speak BGP with a Cisco IOS 12.2 based router.

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

# Why Prioritize Network Traffic?

**BANDWIDTH TO THE INTERNET IS A FINITE RESOURCE AND OFTEN AN EXPENSIVE ONE THIS MONTH WE LOOK AT WHY CLASSIFICATION OF TRAFFIC AND PRIORITIZING THAT TRAFFIC TO MAINTAIN QUALITY OF SERVICE IS CRITICAL TO DAILY BUSINESS OPERATIONS AS SERVICES CONVERGE ON IP DATA NETWORKS...**

By John Buswell

This month we look at convergence and the new capabilities that data networks must provide in order for vital business operations now and in the future. The key to managing the convergence of voice, telephony, video, data and other services is to classify and prioritize network traffic successfully.

### CONVERGENCE

Technological convergence is an on-going challenge in todays data networks. Convergence is a term commonly used in reference to the combination of voice, telephony, data and video services onto a single network. Convergence can reduce costs, provide new features and seamless integrate data. However, as more real-time applications such as video, voice and telephony are placed onto the network, it becomes increasingly important to prioritize network traffic.

### BANDWIDTH

All data networks have a finite amount of bandwidth available. If you plan to add voice and telephony services to the data network, you must also plan to increase the bandwidth available to the network. A business that migrates its telephone services to a Voice over IP system perhaps through a third party such as Vonage, is still utilizing their bandwidth used for data communications over the Internet. Too much voice traffic without increasing the bandwidth will degrade the data services used within the company. If no priority is given to voice communications though, routers will simply attempt to deliver on a first come first serve method of delivery. While it doesn't matter if there is a 30 second pause during a download, it does matter if there is a 30 second pause during a conference call with an important client. Without having a mechanism on the network to classify and prioritize this type of information, you are leaving service quality to chance.

As more services are converged onto the data network in the future, it is vital that your data network has the necessary framework to prioritize network traffic. Without that framework, as more services are converged, you may find resources have not been allocated to increase bandwidth or deploy new equipment vital to business operations in the future.

### PROBLEMS

There are a number of problems that effect todays data networks these include data loss, delay, jitter, out of order delivery, errors and external problems such as Denial of Service attacks.

The edge of your data network refers typically to the equipment between your network and the Internet. This typically involves some kind of router. A router is simply a special computer that takes network traffic, and forwards it to the correct location. You maybe familiar with terms such as firewalls. A firewall is a set of conditions or rules that are applied to the network traffic to determine if it is traffic that is permitted or not. In most cases the highest number of convergence and bandwidth problems will involve traffic entering and leaving your network through this edge point.

These routers have a finite amount of memory, when data comes in at a rate far greater than the bandwidth can support, that data is placed in a queue. The best way to think of it is a queue at a bus stop. The bus comes along, a finite number of people are crammed onto the bus, and it leaves. A number of people are left at the bus stop waiting for the next bus. If the queue or wait for the bus is too long, then people start to leave the queue and try another method such as a taxi. Routers work in a similar manner, if the queue waiting to send data out becomes too long, and the router starts to run out of memory, the router will selectively drop information. The router will discard data indiscriminately, it doesn't care if the information is part of an important phone call or someone in the sales department checking their stock portfolio.

If the router doesn't run out of queue memory, but the bandwidth is severely over utilized, you will get delay. Using our bus analogy, it would be similar to the bus running 30-40 minutes late on each run. Obviously, just like dropped packets, delayed packets, especially in real time applications such as telephony is unacceptable.

### JITTER

This is a problem caused by the variation in time it takes for differ packets to pass through the same set of routers across the Internet. In a stream of information, one packet may take a longer amount of time than another. Now if packets take different routes across the Internet, its possible for those packets to arrive out of order at the remote end.

### OUT OF ORDER DELIVERY

Packets are typically small pieces of a large piece of information. Packets vary in size depending on the type of network that data is traveling through. If a piece of information is too big to fit inside a single packet it is broken up into a series of packets. The way the Internet is designed, its possible for those to arrive out of order. In a congested network, the router may drop the out of order packets as the earlier pieces of the sequence do not arrive quickly enough.

### DENIAL OF SERVICE ATTACKS

Denial of Service attacks are a different problem. Unlike the previous problems that we've highlighted, DoS attacks appear at face value to be legitimate network traffic. DoS attacks are designed to use up the resources of the target so that legitimate users cannot use those services. Classifying and prioritizing critical traffic can help maintain some degree of order during a DoS attack, there are also methods of mitigating DoS attacks through a number of different methods. We will discuss DoS attack mitigation in a later article, it is important to note here that classification and prioritizing of network traffic provides a good foundation for such solutions.

### PACKET CLASSIFICATION

The good news is that most modern routers can be instructed as to whats important and critical and whats not. The technical term for this is called packet classification. Successful packet classification involves both business and technical units of a business working closely, to identify whats critical to daily business operations and what is lower priority. Packet classification should start at a top level, which services are important to the business. Typically this will be methods used to communicate with customers and those necessary to conduct business. For most companies this will be Voice, Email, Ecommerce and perhaps Instant Messaging.

Out of these technologies, we will say that voice should have the highest priority, followed by instant messaging. In our example, the company relies heavily on instant message to follow up on Internet based sales leads. The company's e-commerce solutions are equally as important but not quite as sensitive as the voice applications. Customers can handle a few seconds of delay waiting for the pages to load in the event of network congestion. Finally, Email is given the lowest priority as mail servers will retry every 4 hours for at least a few days to attempt delivery in the event of heavy congestion.

Classification then continues, working down outbound traffic such as web usage, and perhaps blocking undesirable services. This classification is then translated into configurations for routers, switches, firewalls and other equipment on the network.

### QOS

You may already have what is known as an Service Level Agreement, or SLA with your Internet Service Provider(s). The SLA is a guarantee of service, indicating the guaranteed level of performance, throughput and latency based on mutually agreed measures. The service provide typically uses QoS based technologies such as prioritizing traffic to ensure the SLA is maintained throughout the life of the contract.

Prior to deploying QoS, it is important that similar internal SLA-like documents are drafted to provide guidelines for what is expected from the network. By creating SLA type documents, it is very easy to determine if network upgrades and additional Internet bandwidth purchases are really necessary to the business.

QoS is typically deployed using DiffServ or differentiated services. In the diffserve model, packets are marked according to the type of service they need. In order to mark these packets, they first must be identified or classified, which is what we discussed in the previous section. The SLA type documents drafted provide easy guidelines for network engineering staff on how to mark the services.

Once the packets are marked, the routers and switches across the network will queue and prioritize the traffic accordingly.

### POLICY ROUTING

Policy routing is a method of looking at packets as they come into the router, and based on attributes those packets have, to forward them along a specific path. In a local network, you may have multiple Internet connections. It might be desirable to route all voice communications over a larger bandwidth connection, or route all traffic to a particular site over one link because it has a short distance to travel. Policy routing is similar to QoS in that packets are classified, optionally marked and then handled differently through the network. Linux has a wide range of features that can be used to provide policy routing.

### TRAFFIC SHAPING

The term traffic shaping refers to the mechanisms used to control the flow of traffic being sent into a network. It is a combination of bandwidth throttling and rate limiting. Traffic shaping can be used to restrict the inbound flow of data on a high-speed interface perhaps at a co-location center where the interface is capable of higher speeds than the rate being paid for. As a result, bandwidth throttle for example can be used to limit the chances that bandwidth overage charges will apply and thus reduce costs.

Traffic shaping can reduce packet loss, provide lower latencies and reduce jitter. Traffic shaping provides a controlled flow of information instead of bursts of information. Instead of having 10 packets in one second, 0 packets in the next second, 10 packets in the next second and so on. The network will transmit 1 packet every 0.2 seconds instead.

### BOTTOM LINE

Daily business operations are relying more and more on high technology Internet based solutions. Many companies utilize services such as Vonage to provide local phone numbers in foreign countries to compete in those markets. As business communications rely on the data networks of today, it is important that networks have the capability to manage and control the traffic flowing across them. As bandwidth is a finite resource, managing that bandwidth efficiently can decrease the number of costly bandwidth upgrades required while insuring that critical services still function correctly.

# RRDtool Demystified

**WANT AN INDUSTRY STANDARD DATA LOGGING AND GRAPHING APPLICATION?**
**WANT TO WRITE CUSTOM WEB BASED MONITORING SCRIPTS?**
**USE RRDTOOL, A TIME-SERIES DATA STORAGE AND DISPLAY SYSTEM.**

By Bharat Shetty

Have you ever wondered how to gather status information from all sorts of things, ranging from the temperature in your office to the number of octets which have passed through the FDDI interface of your router? Gathering the data isn't a big issue, but it is not so trivial to store this data in an efficient and systematic manner. Don't fret. RRDtool lets you log and analyze the data you gather from all kinds of data-sources (DS).

### WHAT IS RRD?

RRD is the abbreviation for Round Robin Database. RRD enables you to store and display time-series data (such as network bandwidth, machine-room temperature, server load average). Data can be stored in a very compact way, and creation of beautiful graphs becomes an easy task. It can be used via simple shell scripts or as a Perl module.

RRDtool is a GNU licensed software developed by Tobias Oetiker, a system manager at the Swiss Federal Institute of Technology. Technically speaking, it is a database. Still, there are some distinct differences between RRDtool databases and other databases:

- RRDtool helps store data; that makes it a back-end tool. The RRDtool command set allows the creation of graphs; that makes it a front-end tool as well. Other databases just store data and cannot create graphs.

- In case of you wonder where the RRD name arises: New data is appended at the bottom of the database table in cases of linear databases. Thus its size keeps increasing, whereas the size of an RRDtool database is determined at creation time. Imagine an RRDtool database as the perimeter of a circle. Data is added along the perimeter. When new data reaches the starting point, it overwrites existing data. This way, the size of an RRDtool database always remains constant.

- Other databases store the values as supplied. RRDtool can be configured to calculate the rate of change from the previous to the current value and store this information instead.

- Other databases are updated when values are supplied. The RRDtool database is structured in such a way that it needs data at predefined time intervals. If it does not get a new value during the interval, it stores an UNKNOWN value for that interval. So, when using the RRDtool database, it is imperative to use scripts that run at regular intervals to ensure a constant data flow to update the RRDtool database.

An associated time stamp is stored and can be assigned for every data update. Time is always expressed in seconds passed since epoch (01-01-1970).

where to obtain it?
Download latest src code package (rrdtool-1.2.12.tar.gz), from this site:

*http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/?M=D*

Building instructions are available here:

*http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/doc/rrdbuild.en.html*

Be sure that you have these libraries installed. Get them from:

*http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/libs/*

data graphing
Create an empty RRD database using rrdtool create.

**rrdtool create** filename [--start|-b start time] [--step|-s step] [DS:ds-name:DST:dst arguments] [RRA:CF:cf arguments]

```
rrdtool create loadav.rrd --step 10
 DS:load:GAUGE:30:0:100 \
 RRA:AVERAGE:0.5:1:9600 \
 RRA:AVERAGE:0.5:4:9600 \
 RRA:AVERAGE:0.5:24:6000
```

Here you are creating a database named loadav.rrd to graph load average on your machine. Step of 10 seconds means that the database has to be updated every 10 seconds. To update you can use a script which will have to run every 10 seconds.

DS (Data Source) is the actual variable which relates to the parameter on the device that has to be monitored. In the example above, load is the Data Source.

DS:variable_name:DST:heartbeat:min:max

You can have as many Data Sources as you want. The Data Source Type [DST] defines the type of the Data Source [ DS]. In this example it has been declared to be of the form GAUGE so that it doesn't save the rate of change. Instead, the actual values themselves are saved. For example, you can plot the memory consumption using this.

The next parameter we will discuss is heartbeat. As you see in the example, we have defined heartbeat to be 30 seconds. That means that if the database doesn't get a primary data point within the 15 secs, it will wait for another 15 secs, 30 seconds in total. If no data is given, an unknown value will be saved into the database.

The next parameters are min and max. They specify the minimum and maximum values of the variable (load) whose values we are storing into the database. Any value which falls out of this range will be marked as unknown.

Now we come to the discussion of Round robin archives [RRA] . You will define a round robin archive using the keyword RRA. The syntax for defining an RRA is as below.

RRA:CF:xff:step:rows

If we see our example, first RRA definition is like this.

RRA:AVERAGE:0.5:1:9600

Here consolidation function [ CF] is AVERAGE. A consolidated data point is averaged. Other consolidation methods allowed are LAST, MAXIMUM, and MINIMUM. Only 1 PDP is averaged to form a CDP. A total of 9600 rows of these CDPs are being archived here. Each PDP shall occur at 15 seconds. Many RRAs can be defined for single database. For example, here 1 or 4 or 5 PDPS can be averaged.

### CREATE SCRIPTS TO WRAP RRDTOOL

You can wrap the rrdtool inside a script (shell / Perl etc.). Let's discuss this example.

```
#!/bin/bash
echo "updating load.."
echo ""
CURLOAD=`cat /proc/loadavg | cut -f 1 -d \ `
rrdtool update loadav.rrd N:$CURLOAD
CURTIMEIS=`date`
echo "updated at "$CURTIMEIS" with "$CURLOAD
echo ""
sleep 10s
```

CURLOAD is a variable which will be used to store the output of the command cat /proc/loadavg | cut -f 1 -d \

cut -f 1 -d \ means remove sections from each line of the file and then output only that field. So the value in /proc/loadavg will be copied into CURLOAD each time you run the script.

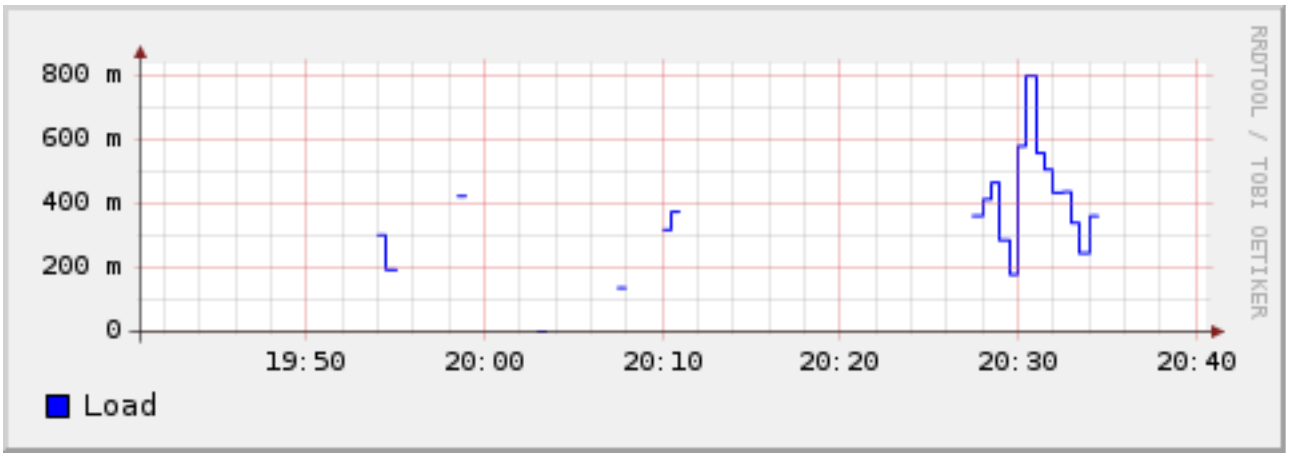
### PLOTTING USING RRDTOOL

RRDtool has the nice feature of generating graphs from the statistics stored in the database. The parameters supplied on the command line are used to generate the graph. A graph can show many data sources.

```
graph loadav.png \
DEF:load=loadav.rrd:load:AVERAGE \
LINE1:load#0000ff:Load --start -1h
```

Different variables can be presented in five different shapes in a graph - AREA, LINE1, LINE2, LINE3, and STACK. AREA is usually represented by a solid colored area with values as the boundary of this area. LINE1/2/3 (increasing width) are just plain lines representing the values. STACK is also an area but it is “STACKed” on top AREA or LINE1/2/3. The plotting takes place in the order in which variables have been defined in graph command.

The resulting graph is shown on the next page.




### LIGHTTPD RRDTOOL MODULE

Now we will discuss the lighttpd RRDtool module. This module is a secure, fast, compliant, and very flexible web server that has been optimized for high-performance environments. It has a very low memory footprint compared to other web servers, and takes care of CPU-load. Its advanced feature-set (FastCGI, CGI, Auth, Output-Compression, URL-Rewriting, and many more) make lighttpd the perfect web server software for every server that suffers load problems.

lighttpd is available from the url below, along with the installation instructions.

*http://www.lighttpd.net/download/*

*http://trac.lighttpd.net/trac/wiki/TutorialInstallation*

configuring lighttpd to use rrdtool module
If you compiled from the source, then copy the temporary configuration file from doc directory within the source directory for lighttpd to /etc/lighttpd/. Open lighttpd.conf using an editor like vim or emacs. Now carefully examine the configuration file. Observe the section named server.modules :

*server.modules = (*
*# "mod_rewrite",*
*# "mod_redirect",*
*# "mod_alias",*
*...*
*...*
*# "mod_rrdtool",*
*# "mod_accesslog")*

Please note that the mod_rrdtool has been uncommented here. This uncommenting will allow us to use the rrdtool.
Next observe the server.document.root. Make sure it points to the correct directory /var/www/pages

Next, set logging as below:

*server.errorlog =*
*"/var/www/logs/lighttpd.error.log"*

Set the port on which lighttpd should run on your system.

*server.port = 3000*

Finally, set the rrdtool path and the round robin database archive that you will use to plot the graphs in the configuration file.

*rrdtool.binary = "/usr/bin/rrdtool"*
*rrdtool.db-name = "/var/www/pages/lighttpd.rrd"*

Now you are ready to run the server. First, check that your config is okay:

*$ lighttpd -t -f lighttpd.conf*

If it is okay, you will get a SYNTAX OK message. Now start the server for testing:

*$ lighttpd -D -f lighttpd.conf*

and point your browser to

*http://127.0.0.1:3000/*

To stop the server again, just press ctrl-c.

### GENERATING GRAPHS

Creating the database:

```
rrdtool create lighttpd.rrd --step 10 \
DS:load:GAUGE:30:0:100 \
RRA:AVERAGE:0.5:1:9600 \
RRA:AVERAGE:0.5:4:9600 \
RRA:AVERAGE:0.5:24:6000
```

This will create the database in the directory that has been specified in the configuration file. Next we will update the values into the database using this shell script. (update.sh)

```
#!/bin/bash
echo "updating load.."
echo ""
CURLOAD=`cat /proc/loadavg | cut -f 1 -d \ `
rrdtool update loadav.rrd N:$CURLOAD
CURTIMEIS=`date`
echo "updated at "$CURTIMEIS" with "$CURLOAD
echo ""
sleep 10s
```

Finally we need to generate the graph, so we will create another shell script rrd.sh as shown below.

```
#!/bin/sh
RRDTOOL=/usr/bin/rrdtool
OUTDIR=/var/www/pages/
INFILE=/var/www/pages/lighttpd.rrd
DISP=" DEF:load=$INFILE:load:AVERAGE \
LINE1:load#a0a0a0:Load "

$RRDTOOL graph $OUTDIR/loadav.png --start -
1h $DISP

$RRDTOOL graph $OUTDIR/loadav.png --start -
2h $DISP
```

This shell script is simple. The RRDTOOL variable has the path to the rrdtool binary on the system. OUTDIR holds the path where the PNG (image of the graph ) will be generated. The database path has been assigned to INFILE. DISP will hold the rrdtool graph definition parameters. We will them plot the graph using the rrdtool graph command.

Run update.sh for few minutes. Then run rrd.sh. This should create a PNG of the graph in the path defined in OUTDIR that is /var/www/pages. Next we will create HTML to display on our web server ( lighttpd server ).

```
<HTML>
<HEAD>
<TITLE>Load Average Graph</TITLE>
</HEAD>
<BODY>
<H1>Load Average Graph</H1>
<IMG src="loadav.png" alt="Load Average">
</BODY></HTML>
```

Fire up your browser and type:

*http://localhost:3000/index.html*

As you'll see from the results, you are looking at the real time load average statistics on the machine itself. You can also monitor other metrics such as CPU usage and memory usage. So this is how we use the RRDtool lighttpd module.

### CACTI: A COMPLETE RRDTOOL-BASED GRAPHING SYSTEM

Cacti is a complete network graphing solution designed to harness the power of data storage and graphing functionalities provided by RRDtool. Cacti provides a fast poller, advanced graph template capability, multiple data acquisition methods, and easy user management.

### CACTI: DATA SOURCES

The paths to any external script/command, along with any data that the user will need to "fill in" can be fed to cacti for the purpose of data gathering. A cron job gathering of this data and subsequently population of the MySQL database/ Round Robin Db will happen.

### CACTI: GRAPHS

Once one or more data sources are defined, a RRDtool graph can be created using the data, all of the standard RRDtool graph types, and consolidation functions. A color selection area and an automatic text padding function also aid in the creation of graphs to make the process easier. It is sort of more robust front end management solution for RRDtool which makes the task of RRDtool users easier. Some of the ways to display the graphs are available such as standard "list view" and a "preview mode," which resembles the RRDtool fronted, "tree view" (which allows you to put graphs onto a hierarchical tree for organizational purposes).

### CACTI: USER MANAGEMENT

A user based management tool is built in so that you can add users and give them rights to

certain areas of cacti. You can create users who can change graph parameters, while allowing others to only view graphs. Each user shall also maintains their own settings for the viewing of graphs.

### CACTI: TEMPLATING SCALABILITY

Lastly, cacti is able to scale to a large number of data sources and graphs through the use of templates. This allows the creation of a single graph or data source template which defines any graph or data source associated with it. Host templates enable you to define the capabilities of a host so cacti can poll it for information upon the addition of a new host.

For more information please visit *http://cacti.net*

### CONCLUSION

RRDtool is an effective, robust, and seamless solution for graphing nearly every imaginable chore on our systems, LAN networks, and servers. We have also illustrated, using simple examples, how to collect data in Round Robin database to produce a graph for monitoring the load average. This can be extended to monitor other parameters such as CPU usage and memory usage.



***Bharat Shetty, aged 23 years is a software engineer by profession. He is a supporter of free software movement and is active member of several LUGs in India. He has remained a GNU/Linux enthusiast and hobbyist since he started engineering studies computer science while at SJCE, Mysore, India. He is very passionate about programming and during his freetime he likes to go on treks and shoot pictures. Other passions includes reading books, writing etc. Bharat works for IBM India.***

# (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 //



# Prioritizing Voice Communication

**QUALITY OF SERVICE (QOS) CAN BE USED TO PRIORITIZE VOICE TRAFFIC OVER BUSY IP DATA NETWORKS. MUHAMMAD HAMMAD LOOKS AT THE BEST METHODS FOR PRIORITIZING VOICE TRAFFIC AND THE CAPABILITIES WITHIN THE LINUX KERNEL...**

By Muhammad Hammad

Quality of Service (QoS) has emerged in the field of networking in the last decade or so as a relative term. It means "as a client, how satisfied are you with quality of service of the network/Internet?".

QoS can be measured in a variety of parameters, such as availability, bandwidth, loss, and latency. QoS is often related to the capability of the network to provide better service to the selected traffic type. Such diverse types of traffic is possible because of the flexibility of IP networks, and end-to-end transport protocols running over IP, while IP itself provides best-effort services and does not guarantee any QoS. For one user, QoS may refer to the smoothness of video playback, while at the same time, and over the same network, web browsing could be more important for another user. Such types of traffic have completely different sets of requirements, in terms of acceptable parameters for loss, latency etc., and yet both users are on the same network and require QoS.

It is sometimes argued that one should aim to increase bandwidth, as opposed to deploying complex QoS solutions, but that's really a separate debate. The important question here is "how QoS can be achieved anyway?"

QoS can be measured primarily in terms of network availability, bandwidth, delay, jitter, and loss. There could be other parameters as well. For instance, ATM specifies peak-to-peak cell delay variation, cell loss ratio, maximum cell transfer delay, etc. We first need to identify the types of network traffic, and then identify the behavior and requirements imposed by each type of traffic.

QoS architecture is rather complex, and employs a number of different techniques, such as identification and marking techniques, queuing, scheduling, traffic shaping and policing, congestion management and avoidance, and so on. End-to-end QoS levels can be categorized as follows:

### BEST-EFFORT SERVICE

This is a general purpose service model suitable for applications such as email and file transfer. The application sends data without any agreement in any quantity, and the network delivers the data without guaranteeing any service quality.

### DIFFERENTIATED SERVICE

This service model is used to meet the desirable QoS functionalities. Each network device tries to provide the requested QoS behavior based on specifications in each packet. The specifications can be made, for example, by setting type of service (TOS) octet, now called as differentiated services field (DS), in the IP header. Using these QoS specifications, the intermediate network device will be able to classify the packets and provide the desired level of services.

### INTEGRATED SERVICE

This is similar to the differentiated service model, but here the application explicitly notifies the network devices of its traffic profile- that is, it requests a specific kind of service. The application will send data only after the request is confirmed from the network.

### VOIP

VoIP is a real-time application and is extremely sensitive to delay and loss. Running voice over traditional telephone networks does not create problems because the total bandwidth of the network is dedicated to voice traffic. But, for VoIP, delay, loss, and jitter play a significant role in quality of voice transmission, because of other traffic running on IP. How can we achieve quality voice transmission over an IP network? The solution lies in prioritizing voice traffic and requesting that the intermediate network devices give it preference. The process of prioritizing voice traffic involves several key steps. A brief overview of is given below:

Before applying QoS mechanisms, we need first

to classify the type of traffic. Classification is the process of identifying the packets and grouping them on the basis of their behavior. For example, for VoIP traffic, a network device must first identify it. This can be done is several ways;

- at layer 4: using source and destination port numbers.
- at layer 3: using source and destination IP

The above methods of classifying packets are done on a per-hop basis, as every intermediate network device has to perform the identification method to identify packets. A more efficient and simpler technique is to mark the packets for network-wide use. This can be achieved by setting the type of service (TOS) in the IP header. TOS field is one byte in length, and its three most significant bits are called an IP precedence. IP precedence defines eight possible values that can be used for the desired quality of service. Differentiated services (DS) architecture introduces DS field, which supersedes the existing TOS and defines a differentiated services code point (DSCP). DSCP uses the first six bits of TOS field, and thus now 64 DS classes can be defined. The remaining two bits in TOS are unused at the moment.

The first three bits in DSCP designate the class selector and are compatible with IP precedence. DSCP classes include best effort, assured forwarding 1, assured forwarding 2, assured forwarding 3, assured forwarding 4, and expedited forwarding. Expedited forwarding provides low-latency and high priority services, and is recommended for VoIP. Once the VoIP traffic has been classified, each intermediate network device can then apply QoS features to achieve the desired quality of voice communication.

### QOS QUEUING

Another important QoS function is queuing. When congestion occurs, queuing all packets in a single queue will not be the optimum solution. Congestion management deals with this situation and uses queuing algorithms to sort out the traffic and then service it using some prioritization techniques. There are many different queuing algorithms, each with its own unique characteristics. Algorithms include First-in, first-out (FIFO), Priority queuing (PQ), and Custom queuing (CQ).

PQ is required for VoIP, and the recommended PQ for VoIP is Low latency queuing. Queuing mechanisms are able to distinguish traffic based on the classification technique discussed earlier. A packet is classified at the edge of the network so that the intermediate routers, along the path, identify and process them accordingly.

### LINUX KERNEL SUPPORT FOR QOS

The Linux kernel includes options for configuring QoS on interfaces. The QoS options can be enabled in the kernel in "Networking-> Networking Support -> Networking Options -> QoS and/or fair queuing". If "QoS options" is disabled, the kernel will, by default, choose FIFO scheme for queuing. The Linux kernel 2.6.15 provides plenty of queuing algorithms and classifications including class based queuing, random early detection , token bucket filter, etc. Moreover, it also supports resource reservation protocol (RSVP), classifying packets according to netfilter marks, a network emulator to simulate WAN conditions (loss, delay etc.), rate estimator to estimate rate-of-flow for network devices and queues, etc. QoS options can be handled by using iproute2 suite, which includes ip and tc for TCP/IP configuration and traffic control respectively. Kernels, prior to 2.6.15, also include QoS options but the hierarchy may be slightly different from the one mentioned above.

Before we see an example on how to prioritize VOIP using Linux QoS mechanisms, we need to first understand some important concepts of Linux traffic control. Linux traffic controller has the following important components:

### QDISC

A queuing discipline or a scheduler that defines the behavior of queuing- i.e., how to queue the packets and which packet to serve first. For instance, FIFO queuing treats the packets on a "first come, first served" basis.

### CLASS

qdisc can be classified into classful qdisc and classless qdisc. A classful qdisc itself is not a queue, but rather associates itself with multiple classes, and each of those class contain a qdisc. In other words, a classful qdisc does not queue packets itself, but rather, it further classifies queues, which are responsible for queing the packets. A class can be tied with a queue, which in turn can define another class. Thus, a hierarchy of classes and qdiscs can be provided to support complex traffic control scenarios. For instance, a priority queuing scheme contains

multiple classes and each class has pfifo queue- three classes of priority, 1 through 3, with 3 being the lowest priority. When a packet arrives, it is put into one of those classes, based on some classification. At the moment, class based queuing (CBQ), hierarchy token bucket (HTB), and priority (PRIO) classful qdiscs are supported.

Classless qdisc, on the other hand, is purely a single queue e.g. FIFO. Linux supports FIFO, random early detection (RED), stochastic fairness queuing (SFQ), and token bucket filter (TBF) classless qdiscs.

### FILTER

Filters are used to select packets, based on some classification, and dispatch them accordingly to one of the associated queues with classful qdisc. For instance, we want a packet coming from a particular source to be given the highest priority in PRIO, and would then use filter to distinguish and forward the packet to highest priority queue.

In the example given below, a sample PRIO queue is created using tc. PRIO is a classful queue and creates three default priority classes, known as bands. Each class itself uses pfifo queuing discipline to store and forward packets. By default, in the absence of any filtering scheme, packets are mapped to a band based on the TOS value.

Each interface has one "root queue" and every class and classful qdisc is identified by a handle. The handle consists of <major:minor> number. All classes having the same parent have the same major number and unique minor number. The root starts with 1: (we don't need to write zero for root).

*# tc qdisc add dev eth0 root handle 1: prio*

This above command creates PRIO qdisc. The PRIO queue creates, by default, three classes, 1:1, 1:2, and 1:3, and each class has pfifo qdisc. The command says "attach qdisc prio to device eth0 and assign it a handler 1:".

Next we need to filter the traffic for a specific port. For example, RTSP (554), to highest priority class i.e., 1:1. By default, the PRIO queue filters the traffic based on TOS- that is, the highest priority is directed to 1:1 (band 0), medium priority to 1:2 (band1), and lowest priority to 1:3 (band 2).

*# tc filter add dev eth0 protocol ip parent 1: prio 1 u32 match ip *
*dport 554 0xffff flowid 1:1*

*# tc filter add dev eth0 parent 1: protocol ip *
*prio 2 u32 match ip dst *
*192.168.100.100/24 match ip dport 80 *
*0xffff flowid 1:2*

The first command above creates a filter on node 1:, assigns it a priority 1 based on the RTSP traffic (destination port 554), and sends it to band 1:1. The second command adds filter that matches the destination IP address and port number, assigns priority 2, and sends it to band 1:2. The hexadecimal number 0xffff defines the pattern mask.

*# tc filter add dev eth0 parent 1: prio 1 *
*protocol ip u32 match ip *
*tos 0x28 0xff flowid 1:1*

Using TOS field we can assign the traffic which requires low latency, e.g. VOIP, to the highest priority queue. The above command shows such capability, and the TOS value used here, 0x28, can be replaced with the desired filtering criteria to prioritize traffic.

The examples shown here are simple yet powerful enough to demonstrate the capabilities of Linux traffic control, which can be used to handle complex traffic control scenarios. In addition, tc can also mark the DS field in the packets, but it requires more deeper understanding of tc and thus cannot be covered in this introduction. Moreover, packets can also be marked using iptables, and tc can take advantage of such a mechanism to filter the traffic.

In a nutshell, Linux provides a complex yet powerful and flexible tool to handle traffic QoS.

### REFERENCES:

RFCs: 1349, 2474, 2475

*http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/qossol/qosvoip.htm*

*http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_1_19/config/qos.htm*

*http://opalsoft.net/qos/*

***Muhammad Hammad, based out of Pakistan is General Manager, Enterprise Data Networking at Spliced Networks LLC.***

# >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

# Deploying Open Source DNS Solutions

**THE DOMAIN NAME SYSTEM (DNS) IS A MISSION CRITICAL SERVICE THAT POWERS THE INTERNET ON A DAILY BASIS. LEARN HOW TO DEPLOY OPEN SOURCE DNS SOLUTIONS IN A SECURE AND OPTIMAL MANNER...**

By John Buswell

The Domain Name System (DNS) provides the human readable name to IP address mapping that is used across the Internet. DNS is a special type of distributed database system that maintains specific information associated with a domain name. The most important piece of information is the IP address associated with a specific host name or the domain name itself. DNS is an important mission critical service for any Internet based business, and provides customer access to all your public Internet services through a human readable name such as www.google.com. Without DNS, if you wanted to use Google, you would have to remember 64.233.167.99, and if that server was down, 64.233.167.104, and so on. Now, add some of your favorite sites, perhaps www.cnn.com, www.linux.com, www.cisco.com and suddenly that's a lot of numbers to remember without DNS.

### DOMAIN REGISTRATION AND TLDS

The Top Level Domain (or TLD) is the rightmost part of the domain name- .com, .net, .org, etc. In today's world, there are a wide range of TLDs, ranging from content-specific such as .museum, to country-specific, such as .ie for Ireland. The first step in getting your business or website on-line is to register a domain. Domains are unique, and typically consist of your business name, brand or some combination depending on what's available. If you have a more generic business name such as Acme Computers Inc, you might find that someone has already registered acmecomputers.com. This could be someone operating a company with the same name in a foreign country, or someone who has simply registered popular name combinations. Once you have decided upon a name, you register it, and provide some basic information about your network, such as contact information, and your DNS server IP addresses. Now at this point you might have a chicken-egg scenario where you want to register the domain prior to rolling out your DNS. Various registrars such as Godaddy.com free DNS services.

### DOMAIN CHECKS

Some domain registries have stricter regulations that others. For example, some country specific registries require that your primary and secondary DNS servers are on separate IP networks. Most will require that you have at least two DNS servers, a primary and secondary server. If you only have one server, a number of companies offer DNS secondary services, where they act as your secondary DNS server. Most ISPs if you are a customer with some kind of business-class service such as T1, will provide secondary DNS services for free or a relatively low cost.

### SERVER OPERATING MODES

There are two modes of operation for DNS servers – Caching and Authoritative. Caching DNS servers take queries from a restricted group of clients (such as an office LAN) and query servers on the Internet for responses. When a response comes back, the caching DNS server will forward the response on to the client. Depending on the TTL (Time to Live) on the response, the caching DNS server will cache the response for that specific amount of time. What this means is that the next time a client requests the same DNS information, if the TTL has not expired, the DNS server will respond with the information within its cache instead of querying the Internet again.

Authoritative servers answer requests from other DNS servers and clients for only the domains for which they are configured as either a master or slave. On our DNS server we are authoritative for o3magazine.com, thus requests for o3magazine.com are answered, but requests for google.com are refused.

Bind 9.x supports an internal / external view solution that allows a DNS server to provide different responses to internal and external clients. This feature can be used to safely run both a caching and authoritative DNS server on

the same system. However, if you have the resources and the capabilities to run multiple DNS servers, running caching and authoritative on different servers is the preferred method.

### COMMON SENSE

While it amazes me that this is often over looked, a small amount of common sense when deploying DNS servers goes a long way. Taking a small network as an example, with two DNS servers, placing each DNS server on separate UPS on different power outlets goes a long way. If a UPS fails, it doesn't take out your entire DNS service. If you have multiple Ethernet switches or hubs on your core network, make sure that each DNS server is plugged into a different one, and so on.

Most DNS servers can operate as both a caching and authoritative server. The best approach is to use separate servers for caching and authoritative requests. Generally it is a good practice to use multiple DNS servers. DNS is a mission critical service- if your caching DNS service goes down, your local servers and workstations that use that DNS service can no longer resolve DNS names such as www.google.com. Unless the users are aware of the IP addresses to use, the network availability is severely degraded. If you have more than one caching DNS server, the likelihood of that occurring is significantly lower.

Likewise, if your authoritative DNS server goes down, outside users, including customers. can no longer get to your services. If either DNS service is compromised, then an attacker can redirect your users and customers to any location they like. This can be used to direct users to “fake” copies of banking or other sites, and used to harvest their login information.

It is very important to keep in mind that your DNS server is a trusted resource. Whether you realize it or not, you implicitly trust your DNS server. Whenever you type www.myonlinebank.com in your browser, you are trusting that your DNS server is sending you to the bank and not to a malicious users web site where you're about to give them your information. This reason alone is sufficient for most businesses to consider deployment of their own local DNS servers over using those provided by their ISP.

### OPEN SOURCE DNS SOLUTIONS

There are a wide range of open source DNS solutions available. As DNS is a trusted resource, you need to be sure that the project you select provides you with security features and is maintained by individuals who are security conscious. For the purpose of this article we are going to focus on BIND 9, available from *http://www.isc.org*. MaraDNS available at *http://www.maradns.org* and djbdns available at *http://cr.yp.to/djbdns.html* are worthy alternatives to bind. In fact djbdns offers a highly secure solution and is well worth a look.

### DEPLOYING BIND

Before you deploy bind on your servers it is important to make sure that those servers are secure in the first place. Tools such as netstat -nap and ps aux will provide you with valuable information as to which ports are open on the system and what processes are running. You should shut down any processes you don't need, and disable any unneeded services and open ports. Then following one of the many security guides available on the Internet for your operating system is generally a good practice.

If your operating system supports a package management system, there is often the temptation to take the easy way out. However building from source has several security advantages when done correctly.

### BUILDING BIND 9.3.2 FROM SOURCE

First, simply get and untar the source. Here we're going to save our source build in our projects/dns directory within our home directory for future reference.

```
mkdir -p ~/projects/dns
cd ~/projects/dns

wget ftp://ftp.isc.org/isc/bind9/9.3.2/bind-9.3.2.tar.gz
```

The following commands, assuming you have GPG installed, are used to check the signature for this release:

```
wget ftp://ftp.isc.org/isc/bind9/9.3.2/bind-9.3.2.tar.gz.asc

wget http://www.isc.org/about/openpgp/pgpkey2004.txt

gpg –import < pgpkey2004.txt
gpg –verify bind-9.3.2.tar.gz.asc
```

Once you are happy with the gpg output, you can untar the source:

*tar zxvf bind-9.3.2.tar.gz*
*cd bind-9.3.2*

We are going to deploy bind in a chroot environment, so we're going to install the source to /usr/local/apps/dns. Instead of doing the default /usr/local installation, here we can keep the installation of BIND separate from the /usr/local tree so we know exactly what bind has installed. This makes it a lot easier to remove at a later date.

We're going to build with –enable-threads which enables thread support. In case you are not familiar with threads, threads is an approach which allows multiple things to happen at what appears to be the same time, allowing each thread to give up control or resume when specific events occur. Typically threads will speed up the system a bit, so that it's not sitting idle waiting for I/O access, and can do something else. The –with-pic build position independent code, and –disable-static prevents the static binaries from being built.

*./configure –prefix=/usr/local/apps/dns –enable-threads –with-pic --disable-static*

You let that run, and when its done simply run :

*make*
*su -* (switches to root)
*make install* (now as root)

We have now built and installed bind. However, we need to create a user and group, and also setup the chroot area. First we will create a group called chdns and a user called chdns:

*mkdir -p /home/chroot*
*groupadd -g 8000 chdns*

*useradd -u 8000 -g chdns -d /home/chroot/chdns -c “DNS” -m chdns*

Now that we have a user and group with a UID/GID of 8000 and a home directory of

/home/chroot/chdns, we are ready to setup the environment:

*cd /home/chroot/chdns*
*mkdir -p dev etc/zones/slave var/run*
*mknod dev/null c 1 3*
*mknod dev/random c 1 8*
*chmod 666 dev/{random,null}*
*cp /etc/localtime etc/*

We've now created some directories needed by bind inside of the chroot area. The mknod commands create the null and random devices, and chmod changes the permissions.

Next we need to enable logging. If you are running syslogd then you need to add :

*-a /home/chroot/chdns/dev/log*

If you are running Fedora, CentOS, RedHat or Mandriva for example, this is typically in /etc/sysconfig/syslog.conf. Simply add it inside the SYSLOGD_OPTIONS:

*SYSLOGD_OPTIONS =”-m 0”*

For syslog-ng users you will want to add the following instead the source src { }; block of your config:

*unix-stream(“/home/chroot/chdns/dev/log”);*

Finally, we need to set some permissions:

*cd /home/chroot*
*chown root .*
*chmod 700 .*
*chown chdns:chdns chdns/*
*chmod 700 chdns/*
*cd chdns/*

When you have configured and setup your zones, you should also run these commands under Linux:

*chown -R chdns:chdns *
*/home/chroot/chdns/etc/zones/slave*

*chown chdns:chdns /home/chroot/chdns/var/run*
*cd /home/chroot/chdns*
*chattr +i etc etc/localtime var*

#### CONFIGURING BIND

The bind configuration will be stored in /home/chroot/chdns/etc/named.conf. For the purpose of this article we will look at the named.conf file in 3 sections – ACLs (Access Control Lists), Options, and Zone Configuration.

configuring access control lists (acls)
For our DNS configuration we will have three ACLs. These can be named what you like, but here we will call them transfer, permitted, and bogon. The first list “transfer” is a list of servers to which we permit the transfer of our domains. Typically the only server you want to list in the

transfer ACL are your secondaries. The “trusted” ACL should be limited to localhost unless you are configuring a caching name server, in which case include just your IP subnets that you wish to permit. Finally, the bogon list contains a list of IP subnets which are known to be not in use or reserved, and which should not be the source of requests. Some of these, however, are RFC 1918 reserved blocks, and since most companies use RFC1918 blocks for private IP addressing, so you will need to modify the bogon list accordingly. You can download the bogon ACL from *http://www.cymru.com/Documents/secure-bind-template.html*.

```
acl “trusted” {
192.168.1.0/24; // local LAN
localhost; // localhost
};
```

As you can see the ACL list has a fairly simple format, it consists of the acl keyword, the name in quotes, entries and brackets.

### CONFIGURING OPTIONS {};

There are many options you can configure; below we have outlined the basics for running a secure authoritative DNS server. The configuration is below and is commented.

```
options {
  directory “/etc/zones”;
  pid-file “/var/run/named.pid”;
  statistics-file “/var/run/named.stats”;
  memstatistics-file “/var/run/named.memstats”;
  dump-file “/var/run/named.dump”;
  zone-statistics yes;
  listen-on { 192.168.100.53; };
  transfer-source { 192.168.100.153; };
  // limit DoS attacks
  notify no;
  // Fast transfers
  transfer-format many-answers;
  max-transfer-time-in 30;
  interface-interval 0;
  allow-transfer {
    transfer; // transfer ACL
  };
  allow-query {
    permitted;
  };
  blackhole {
    bogon;
  };
};
```

The listen-on and transfer-source are useful if you have multiple IP addresses configured on your server. By using different IP addresses for transfers and queries, you make life more difficult for anyone looking to assess how your network is configured, especially if you firewall the DNS services on the transfer IP correctly.

### CONFIGURING ZONES

We are going to use the internal / external views so that the local DNS server can use itself to perform DNS lookups. If you are creating a caching name server, you only need the internal-in view.

```
view "internal-in" in {
  match-clients { trusted; };
  recursion yes;
  additional-from-auth yes;
  additional-from-cache yes;

  zone "." in {
    type hint;
    file "db.cache";
  };

  zone "0.0.127.in-addr.arpa" in {
    type master;
    file "db.127.0.0";
    allow-query { any; };
    allow-transfer { none; };
  };

};

view "external-in" in {
  match-clients { any; };
  recursion no;
  additional-from-auth no;
  additional-from-cache no;

  zone "." in {
    type hint;
    file "db.cache";
  };

  zone “o3magazine.com” {
    type master;
    file “db.o3magazine.com”;
    allow-query { any; };
    allow-transfer { transfer; };
  };

};
```

Now that you have the configuration setup, you

need to create the physical zone files. The db.cache file is created with the dig command:

*dig @a.root-servers.net . ns > db.cache*

The 0.0.127.in-addr.arpa file contains what is known as a reverse DNS. While DNS is typicalling mapping something like www.google.com, reverse DNS is also important, as it maps an IP address to a hostname. The format for reverse DNS is the first three octets of the IP address in reverse with .in-addr.arpa appened to the end. Here the 0.0.127.in-addr.arpa file provides the reverse DNS for 127.0.0.0/8.

```
$TTL 3D
$ORIGIN 0.0.127.in-addr.arpa.
@ IN SOA 0.0.0.in-addr.arpa. root.localhost. (
                        1 ; serial
                        8H ; refresh
                        2H ; retry
                        1W ; expire
                        1D) ; min ttl

                IN NS 127.0.0.1.

1               IN PTR localhost.
```

Here is an example zone file for o3magazine.com:

```
$TTL 3D
$ORIGIN o3magazine.com.
@ IN SOA ns1.eur1.splicednetworks.net.
noc.splicednetworks.com. (
                2006012502 ; serial
                2H ; refresh
                1H ; retry
                1W ; expire
                1D ) ; min ttl

        IN NS ns1.splicednetworks.com
        IN NS ns2.splicednetworks.com
        IN MX 10 smtp1.splicednetworks.com.
        IN MX 20 smtp2.splicednetworks.com.
        IN MX 30 smtp3.splicednetworks.com.

localhost       IN A 127.0.0.1
@               IN A 192.168.1.100
irc             IN A 192.168.20.200
httpd           IN A 192.168.20.105
                IN A 192.168.20.106
wwww            IN CNAME httpd.o3magazine.com.
www             IN CNAME httpd.o3magazine.com.
ww              IN CNAME httpd.o3magazine.com.
w               IN CNAME httpd.o3magazine.com.
w3              IN CNAME httpd.o3magazine.com.
web             IN CNAME httpd.o3magazine.com.
```

The IN SOA () information contains the serial number, refresh, retry, expiration and minimum TTL caching value for the domain. The first two IN NS lines specify the name servers which are authoritative for this domain. The MX lines list the mail servers for receiving mail for this domain. The lower the MX value the higher the priority.

The A records (IN A) specify a direct hostname to IP mapping. The httpd entry has two A records; this will cause the DNS server to perform basic “round robin” (picking one for the first request, the other for the next request and so on). The CNAME entries provide aliases which point to the httpd entry.

#### ADDING SECONDARY NAME SERVERS

Adding a secondary or slave DNS server is relatively straightforward. You would follow the same configuration except that you want to add a masters { }; section to the configuration, and the zone entries are a little different. You do not need to create the individual zone files on the secondary server. The DNS server will do that for you, just make sure the server has permission to write to the appropriate directory.

```
masters “mymasters” {
        192.168.100.30;
};

zone "o3magazine.com" {
        type slave;
        file "slave/o3magazine.com";
        allow-query { any; };
        allow-transfer { none; };
        masters { mymasters; };
};
```

You need to make sure that the transfer-source value on the secondary server is in the “transfer” ACL on the master server.

#### CONFIGURING RDNC

BIND includes a utility called RDNC which allows you to use the command line to admin the DNS server remotely or locally. First you need to generate a set of keys:

*/usr/local/dns/sbin/dnssec-keygen -a hmac-md5 -b 128 -n user rndc*

This will create two files starting with Krndc, one with .private and the other with .key. The line you are interested in is the Key: line in the .private file, it should look something like this:

Key: tYgbq8FfRASqsM0dbiJo3g==

You need to create /etc/rndc.conf :

```
options {
        default-server 192.168.1.53;
        default-key rndc_key;
};

server localhost {
        key rndc_key;
};

key rndc_key {
        algorithm hmac-md5;
        secret “tYgbq8FfRASqsM0dbiJo3g==”;
};
```

Then in named.conf:

```
key rndc_key {
        algorithm hmac-md5;
        secret “tYgbq8FfRASqsM0dbiJo3g==”;
};

controls {inet 192.168.1.53 port 953 allow
{localhost;} keys {rndc_key;};};
```

You must replace 192.168.1.53 with the IP address of your server, but you can now use the rndc command. For example, rndc reload will reload the zone files after you have made a modification to a master zone.

#### STARTING BIND

Now that you have bind built, configured and installed, you can start it. Starting bind in chroot mode is simple:

```
/usr/local/dns/sbin/named -u chdns -t
/home/chroot/chdns -c /etc/named.conf
```

The -u tells named to run as chdns, the -t command tells named to run in chroot and -c tells named where the configuration file is located. One important thing to note: the -c command references the named.conf inside the chroot, so /etc/named.conf is referring to /home/chroot/chdns/etc/named.conf.

You can use netstat -nap to make sure that DNS is running on port 53, and on the appropriate address. The ps aux command can be used to make sure that DNS is running as chdns, and tail -f /var/log/messages will allow you to see any errors that might have occurred if DNS did not start correctly.

#### FIREWALLING TCP DNS

DNS operates on both TCP and UDP port 53. Almost all DNS requests from clients will come in as UDP requests. TCP requests are typically used only for tasks such as zone transfers. However, TCP is also used when the response data size exceeds 512 bytes. There is no difference between the DNS protocol running over TCP and the one running over UDP. While many DNS “experts” will advise you to firewall TCP port 53, this is generally a bad idea. The only real reason for blocking TCP port 53 is to secure zone transfers. However, most modern DNS servers, such as BIND, can be configured to secure zone transfers.

Blocking DNS TCP can cause all sorts of interesting and bizarre problems that are difficult to track down to the actual cause. Blocking TCP port 53 also interferes with the normal operation of the protocol, and that is generally a bad idea. If you still want to block TCP 53, and you are certain you will not issue responses larger than 512 bytes, then think very carefully and do your research before casually blocking TCP port 53.

#### CONCLUSION

This article has walked you through the configuration and deployment of DNS services using Bind, an open source DNS solution that is used by many businesses worldwide. Configuring a secure DNS server isn't too involved, and is well worth the effort.

***John Buswell is Founder and Chief Technology Officer at Spliced Networks LLC.***

# 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.**

## See Us in Action
## LinuxWorld Boston
### April 3-6, 2006

LEVANTA®
www.levanta.com
1.877.LEVANTA

WINNER
LinuxWorld
Most Innovative Hardware Solution

# Linux Systems Management with Intrepid

**THE PAIN AND PRESCRIPTION FOR LINUX SYSTEMS MANAGEMENT**
**A NEW LOOK AT LINUX MANAGEMENT TOOLS REVEAL A LEVEL OF MATURITY AND SOPHISTICATION ON PAR WITH WINDOWS COUNTERPARTS**

By David Dennis

In the past, Microsoft and some industry analysts have claimed that Linux has a higher total cost of ownership (TCO) than Windows, and have cited higher systems management costs as the significant shortcoming for Linux. The line of reasoning is that though the Linux hardware and OS may be cheaper, the cost and complexity of managing Linux systems are more compelling reasons to steer clear.

While it's important to take a look at some of the reportedly common challenges associated with Linux management in the enterprise, there's plenty of "light at the end of the tunnel," so to speak. A recent Enterprise Management Associates study titled "Get the Truth on Linux Management" finds that management tools commercially available today for Linux environments are becoming as sophisticated as what's in use for Windows environments. It concludes that management should not be viewed as a red flag when considering the overall TCO of Linux. The report is available in its entirety, for free download, at *http://www.levanta.com/linuxstudy/*.

So, first, let's take at the propagated pain surrounding anti-Linux management...

### ABUNDANCE OF SERVERS

There are many instances where enterprises are migrating from UNIX to Linux - and in the process are replacing large boxes with an abundance of commodity hardware. Thus, the Linux environment typically consists of many more pieces of hardware to manage. Given the significant savings in hardware and software costs, commodity computing with Linux servers makes sense. However, while the Linux hardware and software cost structures are striking, deploying 10 times as many servers causes a daunting administrative burden. The proliferation of Linux servers causes a problem of abundance - the more servers in use, the more differences and interdependencies.

### LACK OF SOPHISTICATED TOOLS

There's an overall a lack of maturity on the point of system management and configuration management tools. That is to say, Linux isn't on a par yet with what's available for UNIX. Most Linux systems today are administered through a series of scripts and freeware that are very flexible and give good "hands on" control, but that also require significant time for installation and maintenance. Scripting and procedural administration for managing hundreds or even thousands of nearly identical servers is grossly inefficient. Repurposing servers on-the-fly to accommodate changing workloads only increases the nightmare.

### VERSION CONTROL AND CONFLICTS

With Linux, you have a huge number of distributions with varieties within those distributions. In addition, managers often tweak the distributions in ways that are specific to the purpose they're serving. By doing this, they end up with a customized flavor of Linux that increases the challenges associated with version control.

### MONITORING

While there's an increasing amount of support in the Linux kernel for hardware monitoring, it's important to note that, since Linux is developed by members of the Open Source community, there are different groups of developers at different companies. This provides freedom of choice, but can create challenges for the system manager seeking an integrated, centralized monitoring and management system.

### DIASTER RECOVERY ISSUES

When 10 servers replace a single server, the chance of hardware failure increases more than 10 times. IT managers running Linux on commodity servers realize that the hardware doesn't have as many redundant components and plan for hardware failure. Fortunately, there are a number of high-availability solutions for

Linux. Having one Linux server fail in a cluster of 10 servers doesn't impact system uptime, but the system administrator must still reinstall and reconfigure Linux on new replacement hardware. Redundant components on commodity hardware means that the system administrator is faced with disaster recovery of individual servers.

### PATCH MANAGEMENT AND DEPLOYMENT

Most IT managers are faced with managing multiple distributions and configurations from one or more vendors. The great openness and flexibility of Linux can create problems.

### THE LINUX MANAGEMENT PRESCRIPTION -- LEVANTA'S INTERPID M.

Linux maturity continues to evolve and has clearly reached a level of maturity whereby organizations of all sizes can run mission-critical applications with minimal management effort, especially those that utilize sophisticated management tools. One such tool is Levanta's Intrepid M.

The Intrepid M is a turnkey Linux management appliance that utilizes an intuitive interface to deploy, rollback and migrate RPM-based Linux servers (whether running RedHat, SUSE, or Fedora distributions) from a central location - all without the need to install the operating system or applications directly on computers. Intrepid M's diskless approach to provisioning marries change control with data virtualization, delivering dramatically faster and more flexible control of Linux on commodity hardware, racks, blades, boxes, virtual machines, and even mainframes.

The Intrepid M includes ready-to-go templates for a variety of workstations and servers, as well as the open source software needed to deploy them allowing users to create and customize their own templates and add software of their choice to the repository. This "plug-and-play" Linux management solution goes beyond management within the data center and includes 1.4 terabytes of storage space that is used to hold software repositories and rollback information for the managed systems. Running over dual Gigabit Ethernet NICS, the Intrepid M allows line of businesses and/or SMEs to manage their Linux boxes from a central location.

Designed for use by a Linux systems administrator with as little as 1 year of experience, the Intrepid M is extremely easy to install and use. Compared to other provisioning and installation servers such as IBM's CSM (Cluster Server Management) and Red Hat's Kickstart – which are very complicated and require advanced Linux systems administrators to operate them - the Intrepid M can be set-up to start managing systems within an hour.

With built-in storage, the Intrepid M appliance removes the necessity of fitting the software into the ecosystem. Conversely, other Linux systems management software solutions are multi-tier, and have to be hooked up to a shared storage network. They also must be fitted into the current shared storage architecture, for specific purposes or applications.

Also of note, the Intrepid handles VMware virtual machines and Xen hypervisors, making it one of the first management consoles that allows Linux shops to accommodate virtual servers seamlessly.

### THE PROOF IS IN THE FIELD

Approximately 100,000 students at the City University of New York (CUNY) use Blackboard (a web-based e-learning software application that provides online teaching and learning tools) and DegreeWorks (which lets students compare their credits and courses to degree requirements online). The Linux servers that power these applications were difficult to keep running running due to constant server failures and the need for hands-on fixes.

CUNY chose Red Hat Linux running on a single chassis of IBM blade servers to support these applications. Unfortunately, the servers had "laptop-quality" IDE drives installed on each blade and would fail frequently. To fix each failure, CUNY's IT staff had to replace hard drives and provision the blades manually. CUNY owned IBM's provisioning server, CSM (cluster systems management) and should not have had to do fixes manually, but they never used it due to its complexity. Rather than CSM, they decided to use Red Hat's Kickstart installation software for installations and provisioning. With Kickstart, one can create a single file containing the answers to questions normally asked during a Red Hat Linux installation. Again, difficulty in using the software led to frustration. Documentation for the RAID adapter was difficult to obtain. On-board rate adapters made the on-board IDB (intelligent disk backup) drives appear as iSCSI drives to the applications running on the OS. When kickstarting a blade, staff had to walk over to the console or the blade at the appropriate time and load the drivers for the hard drive. To add to the nightmare, blade hard drives frequently burnt out. Over a six-week period, one blade hard drive died each week.

Each server failure cost CUNY about eight hours of one person's labor, a high toll for the six-person CIS group.

After two months of server crashes, Arty Ecock (Manager of VM Enterprise Systems for CUNY Computing and Information Systems) began evaluating server provisioning solutions, including looking at advancements in Red Hat Kickstart and IBM CMS. He chose the Intrepid M for its easy-to-use appliance model and diskless approach to provisioning. Kickstart is not necessary anymore at CUNY. As a blade server needs to be re-provisioned, a template is created on the Intrepid M. As blades fail, CIS simply uses existing templates to re-provision the blade, taking about 10 minutes.


### THE TRUTH REVEALED

A current look at the mature Linux management solutions available today (such as the Intrepid M) run contrary to Microsoft's "Get the Facts" campaign which, among other things, aims to disparage Linux management as more complex and expensive than Windows. The current EMA "Get The Truth on Linux Management" study answers these sentiments by stating that management tools that are commercially available for Linux environments are becoming as sophisticated as what's in production use for Windows environments. From a Total Cost of Ownership (TCO) perspective, the playing field is now even, if not in Linux's favor. Redmond, consider the FUD exposed.




***David Dennis brings more than 10 years of experience in enterprise software, systems management and Internet segments. Prior to Levanta, Dennis served in senior product and technical marketing roles with Centrata, Mercury Interactive and HP, as well as earlier positions with Symantec Corporation and Network General.***

# 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

# Deploying Snort Intrusion Detection Systems

**NAVEEN SHARMA HAS CONTRIBUTED THIS ARTICLE ON DEPLOYING SNORT INTRUSION DETECTION SYSTEMS. NAVEEN WALKS US THROUGH THE DESIGN, INSTALLATION AND CONFIGURATION OF A SNORT BASED INTRUSION DETECTION SYSTEM**

By Naveen Sharma

According to Sun Tzu, in the Art of War, "We should not rely on the likelihood of the enemy not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable".

One of the ways to prepare ourselves is to use an Intrusion Detection System (IDS) and to keep a watch on the traffic flowing through our networks. Intrusion Detection Systems provide early warning of attacks or malicious activity taking place from inside or outside your network. Beforehand information will enable us to prepare, act, and counter-attack (if desired).

Broadly, IDS can be divided into Network Intrusion Detection Systems(NIDS) and Host-based Intrusion Detection Systems (HIDS). NIDS are located at strategic points on the network. They monitor the traffic for malicious activity and take pre-defined action against the exception that was observed/logged. This allows the system admin to take appropriate steps.

NIDS detect attacks by monitoring packets in real time on the network. NIDS matches one or more packets against a database of known "attack signatures," and performs protocol decodes to detect anomalies. These signature databases are updated regularly by the vendors and the open source community.

Figure 1 shows the suggested NIDS sensor placed between your firewall and network. In this configuration, you get all the alerts when attacks are taking place from outside or inside of the network. A second NIDS could be on the DMZ port of your firewall, if you have one, or just before your server cluster. Again, decision of sensor placement may vary from company to company. Figure 2 shows NIDS in action.

HIDS is a different technique, in which HIDS monitors local host for unauthorized changes in critical files, such as configuration files on the local host. Once an exception is detected, it generates an alert by sending email or SMS text messages, by logging the action in a text file, or some other action. Thus, the administrator is alerted whenever there is malicious action on the network and servers. Normally, the enterprise can use a combination of both NIDS and HIDS to harness benefits of both techniques.

Since most of today's networks are switch based, the positioning and number of NIDS sensors is immensely important. One prominent candidate for NIDS sensor placement is immediately after the firewall. Some security professionals recommend placing a NIDS sensor outside as well.

In my view, you are primarily interested in any malicious activity from inside your firewall, and any outside attacks that successfully penetrate your gateway firewall. The additional time required by a security professional to analyze these events will consume that professional's time and may indirectly increase the cost of your IDS deployment over time. However, malicious users do need to gather information prior to attempting an attack on a network. The information gathered by an NIDS sensor on the public side of the firewall, may gather critical information that preludes an attack. Thus, the positioning of a sensor on the public side of the firewall, may provide key pieces of information during an investigation. It will also provide you with useful information on what malicious users are requesting, and what kind of information your network is providing them. However, this type of information may also be successfully gathered through a good logging strategy on your firewall. Thus, the placement of an IDS sensor on the public side of the firewall is something that needs to be considered on a case by case basis.

When several IDSs are placed at strategic points, but managed at a central management station, this arrangement is called Distributed Intrusion Detection System (DIDS). Figure 3 shows host based Intrusion Detection System (HIDS) on a mail server and a web server. HIDS monitors local files for any unauthorized changes.

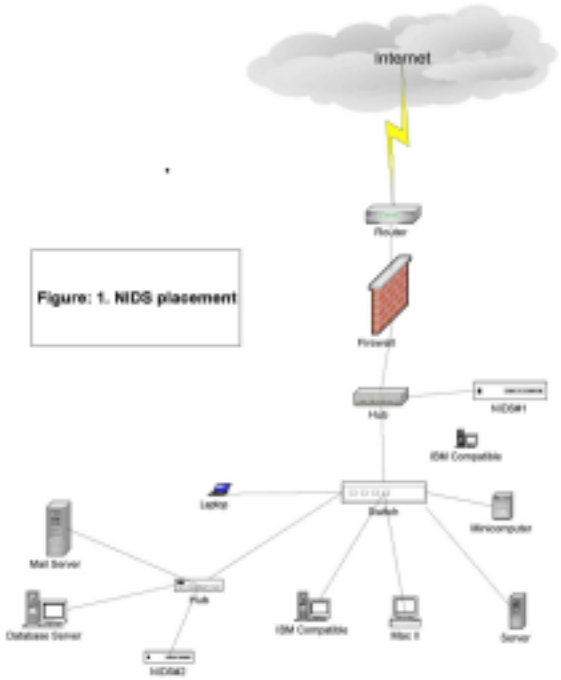



Figure: 1. NIDS placement

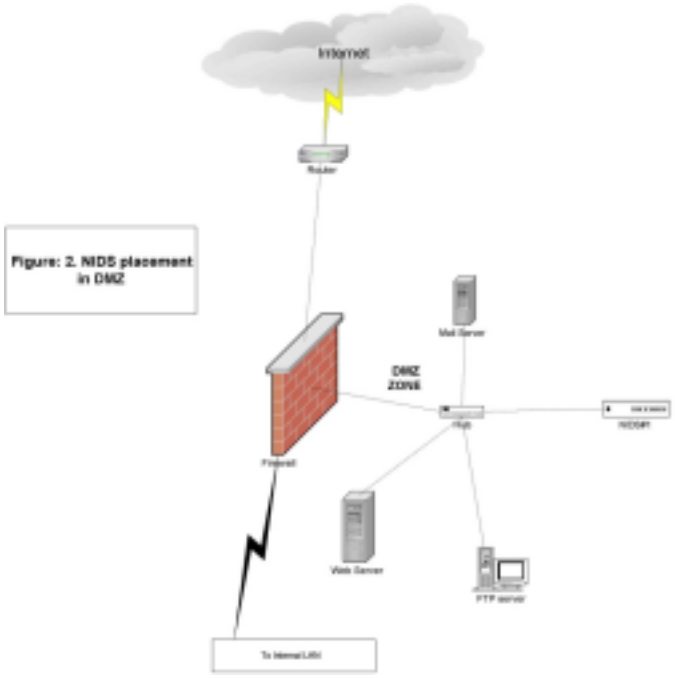



Figure: 2. NIDS placement in DMZ

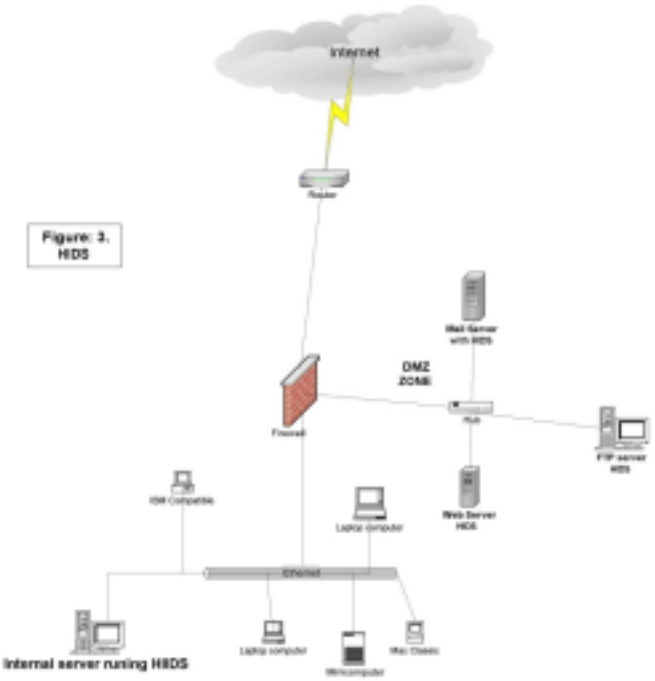
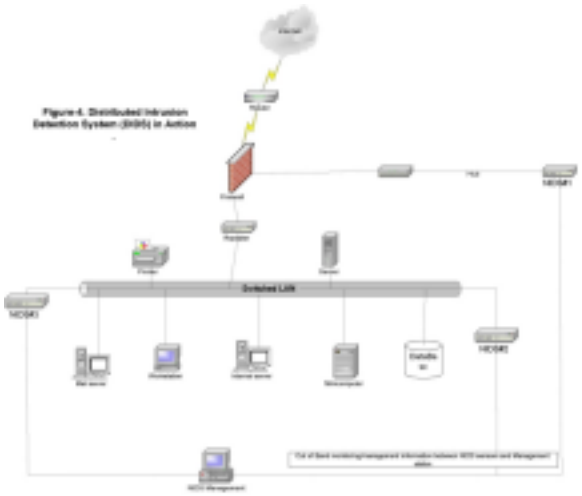



Figure: 3. HIDS



general terms used in context of ids

- A false positive is an intrusion detection error that occurs when a normal activity is mistaken for an attack. This is also called Type 1 error.

- A false negative is an intrusion detection error where an attack is mistaken as normal activity. This is also called a Type 2.

- Signatures are unique data patterns indicating some malicious activity. A signature could point to a virus, for instance, or to an unauthorized attempt to access resources.

- Anomaly detection uses rules or predefined concepts about 'normal' and abnormal system behavior (called heuristics) to distinguish anomalies from system behavior and to monitor, report on, or block anomalies as they occur. Some anomaly detection IDSs implement user profiles. These profiles are based on normal activity, and can be constructed by using statistical sampling, a rule-based approach, or via neural networks.

- Signature detection -- IDS has a database of known attacks, and compares traffic / activity patterns with the known-attack database. If there is a new type of attack not described in its attack database, IDS will note detect this attack and hence no alert will be generated.

Like any other man-made technologies, IDSs also have certain drawbacks, such as the possibility of false positive and false negative errors. Our primary aim is to reduce both of these errors for the efficiency of IDS.

Figure 4 depicts DIDS. There are commercial NIDS products, such as Real Secure from ISS, and Snort from the open source community. Typical HIDS include Tripwire, Samhain and AIDE. In Figure 4, hard lines represent normal network connections, while dotted lines denote out of band communication between NIDS sensors and the NIDS management station. This architecture offers a twofold benefit. First, management and alert information is kept secret from anyone sniffing traffic on the network. Secondly, traffic generated by NIDS sensors and NIDS management does not disturb normal network traffic. All NIDS will send all alerts to one centralized management station. This makes life easier for security administrators.

In this article, I will discuss Snort-based IDS and various features associated with its implementation. Snort is an open source GNU Public License (GPL) Network Intrusion Detection System capable of performing real-time traffic analysis and packet logging. Snort can do protocol analysis, content searching/matching, and can detect attacks and probes such as portscans, CGI, attacks, and spoofing.

I chose Snort because it is free and has an active open source community involved in the continued development of Snort and its rules. Snort is widely deployed in production networks world-wide, protecting more than 100,000 networks.

Snort has three basic modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, Snort reads packets from the wire and displays them in a continuous stream on the console. In packet logger mode, Snort logs the packets to the disk, and they can be examined by TCP dump for later analysis. The most juicy mode is intrusion detection mode, in which Snort analyzes network traffic for matches against predefined rule sets.

One issue with switched networks is that traffic is sent to the intended port only- unlike in hub-based networks, where traffic is visible to every node connected to the medium. If you simply connect IDS to one of the switch ports, then it will not capture anything except broadcast traffic. The work-around for this problem is to use a feature called the Switch Port Analyzer port, or simply SPAN port. SPAN allows the user to copy traffic to this port for analysis or for other purposes. See Figure 5 to see Port SPAN feature in action. Here, port 10 is connected to the internal server (the server under NIDS protection) and is SPAN on port#5. Full traffic on port#10 is copied to port#5 of the switch, and no topology changes are required in performing SPAN.

Alternately, you may connect a small hub just before the server to be monitored as shown in Figure 6. This will require disconnection of the server from network and reconnection to the hub; dedicated network taps are available for both copper and fiber cabling. Taps remain on your network for any connected NIDS or network/Protocol analyzer to monitor traffic.

### SNORT ARCHITECTURE

Understanding Snort architecture will enhance your troubleshooting capabilities. Figure 7 shows Snort architecture. Snort consists of four

components.

1. The Sniffer
2. The Preprocessor
3. The Detection Engine
4. Alerts / Logging

Normally NIC cards are designed to operate in non-promiscuous mode- i.e. The NIC will accept only packets directed to it, and will discard others. NIDS (Snort in this case) forces the NIC to operate in promiscuous mode, thereby allowing the capture of 100% of the traffic on the cable. The sniffer component captures full traffic and passes the packets to the preprocessor module. The preprocessor takes raw packets and checks them against certain plug-ins (like the RPC plug-in). The plug-in checks for certain type of behavior from packets. Once a particular behavior is determined, it is passed on the next component, the Detection Engine. The preprocessor can be enabled or disabled as per our needs.

The detection engine has a rule set (attack signature) which it compares with incoming packets from the preprocessor. If the rule matches the data in the packet, then they are forwarded to the alert processor.

The Alert Processor takes the responsibility of informing the user of the rule-matching either by sending alerts to a log file, using a windows Popup (SMB), or by SNMP trap. Alternately, these alerts can be stored in an SQL database such as MySql or Postgres. There are also third-party software packages for displaying logs and managing Snort rules.

The default location for Snort logs is /var/log/snort. Therefore it is good idea to dedicate separate partitions at the time of installation (perhaps 10 GB or more). You may wish to send these alerts to a centralized syslog server, which can gather alerts from multiple sensors. Swatch can automate the process of sending alerts by email.

### WHICH OS ?

Snort was developed for *nix operating systems and is perfectly married to Linux, Unix, BSD etc. I usually recommend installing Snort on Linux, as it can out perform others in terms of stability, customization and efficiency. Recently, Snort has been ported to the Windows platform as well. Visit *http://www.engagesecurity.com/downloads/#idscenter* for details of IDS Center Snort-based IDS for Windows. On this site you will find fully compiled Snort for Windows and installation is straight forward with executable with documentation.

### INSTALLATION OF SNORT

Download the latest version of Snort from *http://www.snort.org*. At the time of writing, Snort was in 2.X.X version, available for download. I will use 2.X for this article. Save the downloaded Snort to a directory and issue following commands in the same sequence. I am assuming Snort is downloaded to one temporary directory.

```
# tar –zxvf snort – 2.1.0.tar.gz
# cd snort – 2.1.0
# ./configure
# make
# make install
```

That is all. Make changes to your snort.conf file for final configuration described later in this article. Test by issuing the command

```
snort -v
```

With this command, you will display the captured packets.

using snort as a packet sniffer

Snort's greatest power lies in its use as NIDS. But it can also be used to sniff the network and show the packets on console, or to log to a file for later analysis. Issuing a simple command will put the snort into basic sniffing mode and will echo the TCP/IP header to the console:

```
#snort v
```

If you want to see the traffic captured later, the following command will do exactly that, and save the traffic information in /var/log/snort directory:

```
#snort l /var/log/snort h 192.168.1.0/24
```

It will create sub directories under /var/log/snort, one for each IP address.

In this mode you can monitor traffic from specific IPs also. One more advantage you can avail is that Snort formats are read by TCP dump also. One cardinal aim is to show Snort as NIDS, therefore the next section is dedicated to that.

configuring snort as a nids
By default, the Snort configuration file is located in /etc/snort/snort.conf. In this file you are required to tell Snort about your home network, external network address, your local SMTP/POP server, web, DNS server etc. Setting the home network for 192.168.1.0/24 is as simple as adding the following to the configuration:

*var HOME_NET 192.168.1.0/24*

As mentioned earlier, IDS depends upon attack signatures. Similarly, Snort relies upon a series of rules to detect specific types of attacks. Snort installation loads rules from files in the Snort configuration directory, or from a sub directory of it, such as rules. These files have names that end in rules. Snort has documentation available for installation, configuration and implementation, and you can find it at *http://www.snort.org*. If you are really interested full fledged Snort-based sensors, you may download the Snort implementation guide from *http://www.internetsecurityguru.com*. You may write rules for your environment- for example, if you are interested in getting alerts whenever someone tries to make an FTP connection to your accounts server. Snort rule writing can easily be learned, and documentation is available at *http://www.snort.org*.

Third-party tools are available to make your life easier by managing alerts via email, SMS, or through a nice GUI for viewing alerts. Prominent ones are ACID, SnortSnarf and Razor Buck. A Google search might turn up additional tools. Of all of these, I personally like ACID for its nice GUI, and for its features and ease of use.

One last point is to secure your Linux box for implementation. This includes switching off unnecessary services, removing unneeded packages and removing IP addresses from the box so that no connection to sensors will be entertained. Again, you have to get to console to view the alerts in that case. Alternately, you can have two interfaces, one being used for sniffing and connected to the point of interest, and the other one for management of sensors (not connected to the production network). As times passes , new attacks are discovered and rules are made for them and are added to rule set. You should update your rule set in perhaps 2-3 month intervals.

In summary, Snort based IDS can inform you of the health and security of your network. Happy Snorting !!

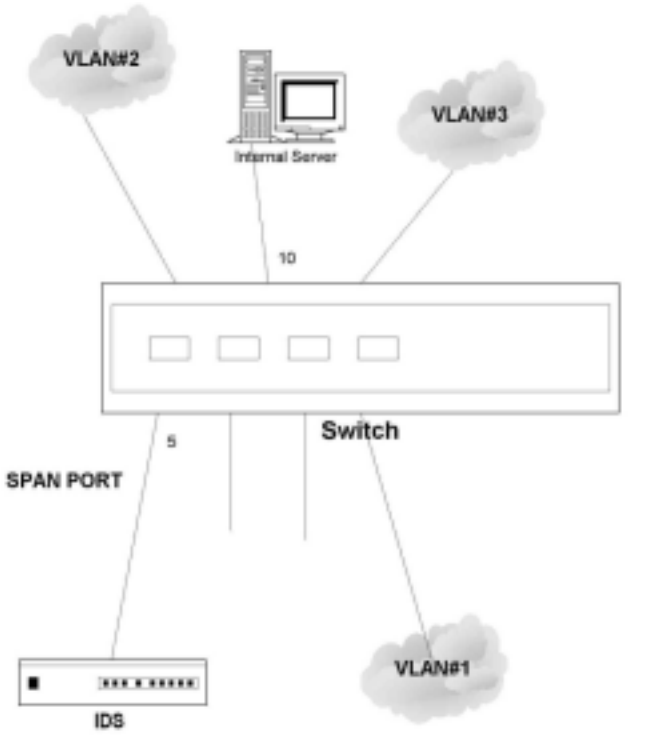



Figure: 5 Port Span in switch

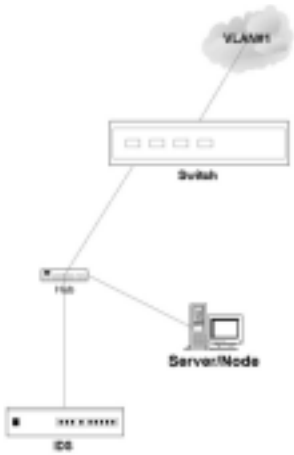


Figure: 6 Placement of hub before server/node for monitoring.


CANADA ON RAILS

# YVR06

DAVID HANSSON
THOMAS FUCHS
DAVE ASTELS
DAVID BLACK
JOE O'BRIEN
JAMES ADAM
STEVEN BAKER
MICHAEL BUFFINGTON
ROBBY RUSSELL
GEOFFREY GROSENBACH
KYLE SHANK
JEREMY VOORHIS
ALEX BUNARDZIC
SEBASTIAN KANTHAK
AMY HOY

## VANCOUVER, BRITISH COLUMBIA

ARPIL 13-14, 2006

TICKETS ON SALE NOW
www.canadaonrails.com