

ISPadmin

by Robert Haskins

Robert D. Haskins is an independent consultant specializing in the Internet Service Provider (ISP) industry.



rhaskins@usenix.org

Managing and Providing ISP Services

Introduction

In this edition of ISPadmin, I look at ways service providers manage and provide mail, Web, dialup, and other types of services to their customers. If I were bringing up any service provider type of business from scratch today, ISPMan (or a similar, usually custom-built package) would be at the core, along with an appropriate billing system.

Background

In most smaller “legacy” ISPs, shell scripts are used to provision services ordered by the customer. These scripts are custom written and driven by the “legacy” billing system. Such provisioning applications tend to take on a life of their own, causing a major maintenance headache for the ISP. For example, each time the provider offers a new type or classification of service, the custom provisioning system must be modified to include this new service. Also, when a new system is added, the provisioning system must be modified to include this new system. When the business model changes, there is a high likelihood that the provisioning system will also need to be changed.

There are many operational issues as well. For example, non-LDAP-enabled systems (smaller providers who use stand-alone files for enabling dialup (RADIUS) accounts) experience a short amount of downtime when the RADIUS service is reloaded after updating the user list. Also, there is always a possibility the service won’t start up correctly when the associated RADIUS daemon is reloaded after adding a user to the stand-alone file.

Most larger service provider operations have resorted to writing custom scripts for this purpose. However, for a smaller provider (and even for larger ones just starting out), an application like ISPMan is a better option. While somewhat constrained by the inherent design in any application, it is written in Perl so the provider can add functionality if need be.

Open Source Solutions

Several open source systems manage and provide ISP services. Vishwakarma, perhaps the oldest service provider management system, doesn’t appear to have been modified in the past three years. It has a lot of the functionality that an ISP would require, such as virtual email, Web-hosting management, and reseller support, and it has an LDAP directory to tie everything together. However, it appears to have several pieces missing. The biggest holes would be lack of a distributed model for back-end services, in addition to quota support. However, it is certainly worth a look. It is released under the GNU GPL.

Another system in the same vein is a package called “vhost” (virtual host), which is also released under the GNU GPL. It has much of the same functionality as ISPMan, with support for virtual email and Web sites. However, its biggest drawback is the lack of an LDAP directory, which would allow a distributed model for enabling back-end services. Hopefully, this is an area of focus for future development. Imagine how the growing ISP would have to throw out the vhost system when they ran out of spare cycles on their existing hardware! It would be much easier, and more cost efficient, to simply add machines to the cluster in order to add capacity.

Commercial Solutions

There are a number of commercial products in this space. Most companies have products in the “virtual server” space as well, which allows a service provider to make (and sell) multiple virtual servers out of one “real” server. (The best known open source virtual server project is vserver.) Many of these commercial products are based upon open source components for their back-end services. Some commercial solutions even manage Microsoft-based products such as Exchange and Windows Server 2000. (As an aside, if you are looking for an open source Exchange replacement, the SAGE members list had a great discussion on the topic back in June 2003.)

The obvious benefit of a commercial solution is the improved support one should get utilizing such a product. Given the level of support available in open source applications such as ISPMan, such differences are decreasing by the day. However, commercial implementations might be useful in certain environments and are certainly worth investigating.

What Is ISPMan?

ISPMan consists of a collection of back-end services (all utilizing an LDAP directory for user information) and a set of custom Perl programs to manage those services. Like the other open source ISP managers, it has been released under the GNU GPL. The major applications ISPMan uses to provide services to end users include:

- Postfix (mail exchange)
- Cyrus IMAP (POP3/IMAP subscriber access)
- Apache 1.x (Web services)
- BIND 8/9 (name services)
- PureFTP (FTP access)
- OpenLDAP 2.x (directory services)
- FreeRADIUS (RADIUS services)
- Horde IMP (Webmail)

The system is scalable, and redundant with appropriate additional hardware. It contains a Web interface for management/provisioning, as well as a command line interface for provisioning accounts via automated mechanisms (e.g., through the service providers billing system) and for other tasks. Any function that can be performed via the Web interface is available via the command line interface on any node in the cluster! It is worthwhile noting that the developers of ISPMan plan on implementing a Simple Object Access Protocol (SOAP) interface for easier integration into the billing system, and other required interfaces as well.

ISPMan has a very nice “reseller” feature, where the ability to add/change/delete services can be delegated to others (e.g., a wholesaler setting up a reseller to sell dialup accounts, Web hosting, etc.). Also, this interface could be used for a large enterprise, where the central IT department gave access to people in each division to add email or dialup accounts, perhaps even under their own subdomains (or completely different domains). This is discussed in “How an Enterprise Might Use ISPMan,” below.

How ISPMan Works

The basis of ISPMan is, of course, the LDAP directory. Where standard attributes don't exist, ISPMan defines its own schema that defines the data structure the ISPMan applications use to communicate between cluster members (e.g., the location of a user's mailbox), and for communications with machines outside the cluster (e.g., DNS data).

A user record, for example, has the following basic attributes (the schema source is in parentheses):

uid (cosine.schema)
uidNumber (nis.schema)
homeDirectory (nis.schema)
userPassword (core.schema)

ISPMan currently makes use of the following schemas (the source is in parentheses):

core.schema (openldap)
cosine.schema (openldap)
nis.schema (openldap)
misc.schema (openldap)
inetorgperson.schema (openldap)
dnszone.schema (from bind9_sdb)
pureftpd.schema (from pureftpd)
ISPMan.schema (ISPMan)
RADIUS-LDAPv3.schema (from FreeRadius)

Each server in the ISPMan cluster runs the “ISPMan-agent” daemon. Its purpose is to communicate tasks to the LDAP server to execute. For example, if a node running the Web user interface manager has a request to add a user, the request gets communicated via the ISPMan-agent interface to the appropriate servers, which, in turn, provision the user on the appropriate servers. Some of the tasks it can perform are the following:

create mailbox
set mailbox quota
delete mailbox
create home directory
delete home directory

ISPMan Installation

As with many projects, the most important piece for a complicated installation like ISPMan is to plan appropriately. Some questions you should ask when planning a project like this include:

- What types of services do I want to offer?
- What equipment can I dedicate to this project?
- How much redundancy do I want?
- What level of high availability am I willing to pay for (99.9%, 99.99%, etc.)?
- How many subscribers do I want the initial system sized for? How many subscribers do I think I will have in six months? In 12 months?
- Which machines will have what function?
- What is the interface into my billing system?
- Are there any other interfaces required for add-on services, such as hosting MS Exchange services?
- Do I have resellers selling my services? What are their requirements?

ISPMan itself is relatively straightforward to install. However, due to the complex nature of these installations, the details are left as an “exercise for the reader.”

Additional components such as antivirus and anti-spam functionality can be added. These can be provisioned as optional additional services in order to build incremental revenue for the provider.

How an Enterprise Might Use ISPMAN

The “holy grail” that ISPMAN solves for the enterprise would be the “single sign on” problem. The single sign on would essentially be the employee’s LDAP directory entry in the ISPMAN application. This would be used for user authentication for every application in the enterprise that required it. It could also be used for a centralized “address book” if so desired.

In addition, ISPMAN could be used to provision and provide email and dialup services for an enterprise. In fact, the reseller capability would be a nifty way to enable divisions of a large organization to add their own accounts, on their own subdomain or even in conjunction with an entirely different domain! The central IT organization would enable IT or division employees to manage their own subset of accounts separate from the rest of the organization.

ISPMAN could be combined with something like Bynari’s InsightServer to provide calendaring, scheduling, contact-list management, etc., for a total enterprise solution at a fraction of the price of MS Exchange or Lotus Notes. The added bonus is that the solution is based on open protocols, utilizing the same “standard” interface (MS Outlook). Bynari even offers an add-on Web client that emulates much of the functionality of the MS Outlook client. While not for every enterprise, certainly the price alone makes it worth looking into.

I wish to thank Atif Ghaffar for his input into this article. Next time, please join me for another installment of ISPadmin. In the meantime, I look forward to hearing your feedback!

REFERENCES

- Amavis: <http://www.amavis.org/>
- Apache: <http://httpd.apache.org/>
- Bynari InsightServer: <http://www.bynari.net/>
- Clam Anti-Virus: <http://clamav.elektropro.com/>
- Cyrus IMAP server: <http://asg.Web.cmu.edu/cyrus/imapd/>
- Ensim: <http://www.ensim.com/>
- FreeRADIUS: <http://www.freeradius.org/>
- Horde IMP: <http://www.horde.org/imp/>
- H-sphere: http://www.psoft.net/h_sphere2_info.html
- Introduction to ISPMAN written by the author, Atif Ghaffar: <http://www.linuxfocus.org/English/September2000/article173.shtml>
- ISC BIND: <http://www.isc.org/products/BIND/>
- ISPMAN home page: <http://www.ISPMAN.org/>
- Lotus Notes: <http://www.lotus.com/>
- Managing ISPMAN’s logs: <http://nakedape.cc/wiki/index.cgi/ISPMANLogStats>
- MS Exchange: <http://www.microsoft.com/exchange/>
- MS Outlook: <http://www.microsoft.com/office/outlook/default.asp>
- OpenAntiVirus: <http://www.openantivirus.org/>
- OpenLDAP: <http://www.openldap.org/>
- Postfix: <http://www.postfix.org/>
- ProFTPD: <http://proftpd.linux.co.uk/>
- PureFTPD: <http://www.pureftpd.org/>
- SAGE members list archive: <http://sageweb.sage.org/resources/maillarchive/sage-members-archive/>
- Neil Schneider’s write-up on installing ISPMAN: <http://www.linuxgeek.net/ISPMAN/>
- SOAP: <http://www.w3.org/TR/SOAP/>
- SpamAssassin: <http://au.spamassassin.org/>
- Sphera: <http://www.sphera.com/>
- vhost: <http://www.chaogic.com/vhost/>
- Virtuozzo: <http://www.sw-soft.com/>
- Vishwakarma: <http://www.kandalaya.com/vishwakarma.shtml>
- vserver: http://www.solucorp.qc.ca/miscprj/s_context.hc