Slide 1

Playing Together

Marty Pauley <`marty@kasei.com`>

火星

O'Reilly Open Source Convention, 2003.

Contents

1 Introduction **2**
1.1 Motivation . . . 3
1.2 Audience . . . 4
1.3 Example Small Network . . . 5
1.4 A Plan . . . 6
1.5 Axioms . . . 7
1.6 Outline . . . 8
1.7 Let's go . . . 9

2 Basic network configuration **10**
2.1 DHCP . . . 13
2.2 DNS . . . 18

3 Intranet web server **24**
3.1 Apache . . . 25
3.2 Knowledge sharing . . . 26
3.3 webmin / usermin . . . 29

4 Sharing **30**
4.1 Printing . . . 32
4.2 File Sharing . . . 34
4.3 MacOS and NetAtalk . . . 35
4.4 Windows and Samba . . . 38
4.5 Users and passwords . . . 46

5 Email **47**
5.1 SPAM . . . 53
5.2 POP and IMAP . . . 54
5.3 Receiving email . . . 56

6 Conclusion **57**

Slide 2

1 Introduction

This tutorial will demonstrate and explain how you can use a GNU/Linux server to provide core network services to a small network of Windows, MacOS, and Unix(ish) systems.

Slide 3

1.1 Motivation

Many small companies have heterogeneous computer systems that they would like to work together, but they are put off by perceived complexity and cost, especially after talking to some vendors.

This tutorial will show that a Free Software system on a GNU/Linux server can do more work than proprietary servers for a fraction of the cost.

Slide 4

1.2 Audience

The tutorial is aimed at novice GNU/Linux system administrators, and experienced NT systems administrators who are moving to GNU/Linux.

Anyone planning to add network services to a small network comprising Windows, MacOS, and GNU/Linux systems.

Slide 5

1.3 Example Small Network

- 2 Windows 2000 PCs
- 1 Windows XP PC
- 1 Windows XP laptop
- 5 MacOS 9 PowerPCs
- 1 GNU/Linux laptop

Slide 6

1.4 A Plan

For some of the services we will examine, there are multiple Free Software solutions. I won't cover all of them.

If my chosen solution is not the most common choice, I will describe the popular solution and explain why I avoided it. Sometimes the popular programs are not the best.

Slide 7

1.5 Axioms

- I believe that Free Software installation is trivial. If you disagree, upgrade to Debian GNU/Linux. All the demonstrations in this tutorial will use Debian.
- I believe that a simple solution is better than a perfect solution.

Slide 8

1.6 Outline

1. Basic network configuration
2. Intranet web server
3. File and printer sharing
4. Internal email system

Slide 9

1.7 Let’s go

So, you’ve installed GNU/Linux on a machine, connected it to your network, and checked that it works.

Give it a name: I’ll call it *ichiban* 一番

Slide 10

2 Basic network configuration

We will assume that our simple network is a LAN that is connected to the Net through a separate router.

The GNU/Linux server can act as a router, but we won't cover that (unless someone asks now...).

Slide 11

Required information

Each network machine needs an IP address. It also needs to know:

- the size of local network
- how to contact the rest of the world
- who to ask for more info

Slide 12

IP addresses

For a small internal network, use private IP addresses.

- 10.*.*.*
- 172.16.*.*—172.18.*.*
- 192.168.*.*

Slide 13

2.1 DHCP

The **Dynamic Host Configuration Protocol** enables your networked machine to configure themselves.

You need a DHCP server to provide the information to the client machines.

Slide 14

ISC DHCP

The standard DHCP server is the Internet Software Consortium dhcpd, but I don’t use it.

It has all the features you will ever need, most of which you will never need in a small network.

It requires “Netlink” devices and socket filtering in your kernel (Networking options).

Using these features seems to make the dhcpd run outside the reach of iptables.

Slide 15

udhcpd

(the **u** should be a μ)

A small and simple DHCP server, perfect for a small and simple solution.

Slide 16

udhcpd.conf

```
interface       eth0
start           192.168.144.101
end             192.168.144.199
opt subnet      255.255.255.0
opt broadcast   192.168.144.255
opt dns         192.168.144.1
opt wins        192.168.144.1
opt router      192.168.144.254
opt domain      oscon
```

Slide 17

/etc/hosts

ichiban has a fixed IP address, so make sure it knows.

```
192.168.144.1      ichiban.oscon ichiban
```

Slide 18

2.2 DNS

The **Domain Name System** is like a telephone directory: you can lookup a name to get a IP number.

Slide 19

BIND

BIND (Berkeley Internet Name Domain), also from the Internet Software Consortium, is the most popular DNS server. Again, it has all the features you will ever need, most of which you will never need in a small network.

If you ever need to use BIND, use version 9; previous versions (4 and 8) have had security problems.

Slide 20

dnsmasq

Another simple solution.

> **Dnsmasq** is a lightweight, easy to configure DNS forwarder designed to provide DNS services to a small network where using BIND would be overkill. It can have its upstream DNS servers automatically configured by PPP or DHCP and it can serve the names of local machines which are not in the global DNS.

Slide 21

dnsmasq configuration

(This page intentionally left blank.)

I told you it was simple!

Slide 22

dnsmasq.conf

I do have one line of config:

```
resolv-file=/etc/resolv.forw
```

I want *ichiban* to use the cache too, so I have `nameserver 127.0.0.1` in `resolv.conf` and the upstream DNS server in `resolv.forw`

Slide 23

Other DNS servers

If you don’t like dnsmasq, you might want to try **pdnsd** (Proxy DNS) or part of **tinyDNS**.

Slide 24

3 Intranet web server

It's a good place to put documentation, so the users know how to use their new services.

Also a great place for communication enhancing applications.

Slide 25

3.1 Apache

Apache is probably the most successful Free Software program in the world. It has over 60% of the market.

Apache 2 has improved internals with similar config.

Either version will be fine. *ichiban* will use Apache 2.

Slide 26

3.2 Knowledge sharing

A wiki can be really useful for sharing knowledge like:

- contact list, internal and external
- HowTos; problem descriptions and solutions
- local interest, food shops!
- other useful information

Slide 27

CGI::Kwiki

There are lots of programs that can create a wiki with a wide range of features.

CGI::Kwiki is a new Perl program that can help you make a wiki. It's simple.

```
cd /my/cgi/directory
kwiki-install
```

Slide 28

Internal weblog

It can be very useful to record your daily activity in a web log.

CGI::Kwiki can be used as a blog for our demo.

You might want to use something else. **Movable Type** is popular, although it’s not Free. **blosxom** is Free, but isn’t pretty.

Slide 29

3.3 webmin / usermin

Two related programs that provide a web-based interface to system admin functions using their own custom web server.

As a minimum, install usermin and its “changepass” module, to let the users change their password. You will need this later!

Slide 30

4 Sharing

Sharing resources, like printers and disk space, is one of the most popular reasons for using a server like *ichiban*.

Different systems don't like talking to each other. In fact, DOS-based Windows systems don't like talking to NT based Windows systems!

Sharing between different systems can be complicated. We'll try to keep it simple!

Slide 31

Peer pressure

The sharing protocols on Windows and Mac systems are **Peer-to-peer**.

Many years ago this was considered a great idea: you didn't need an expensive server, or an expensive system administrator! *Client-server technology was outdated anyway.*

Client-server sharing is actually simpler than peer-to-peer.

4.1 Printing

Slide 32

Samba and Netatalk (you'll meet them soon) provide network printing services, but we'll try another way first: use the **Internet Printing Protocol**.

IPP is supported by recent version of Windows, MacOS, and GNU/Linux.

CUPS

Slide 33

The **Common Unix Printing System** implements the IPP.

IPP is a variant of HTTP, using a different port.

CUPS is configured with a web interface, so let's add some printers.

Slide 34

4.2 File Sharing

In addition to the user home directories, we want to share a common directory.

```
mkdir /home/commons
chmod a+rwx /home/commons
chmod o+t /home/commons
```

Slide 35

4.3 MacOS and NetAtalk

Apple Macs communicate using AppleTalk. **NetAtalk** makes GNU speak AppleTalk.

You might need kernel with “AppleTalk protocol support” (Networking options) enabled. Also works over TCP/IP.

Slide 36

What’s a resource fork?

MacOS files are not like other files. They have two parts: the **data fork** and the **resource fork**. Other systems (like Windows and GNU) only care about the data fork. But MacOS needs both.

MacOS folders also need extra information. Read-only folders need to be partially writable!

Slide 37

AppleVolumes.default

```
~/               "Home"
/home/commons    "Commons"
/usr/share/doc   "Docs"
    options:ro dbpath:/tmp/atalk/doc
```

Slide 38

4.4 Windows and Samba

Windows systems (try to) communicate with SMB, or now CIFS. **Samba** makes GNU dance.

Samba is amazingly powerful. It's much more flexible than any Windows server. Latest Samba can replace Windows domain controllers.

We could have an entire tutorial just on Samba, but we won't.

Let's keep it simple.

Slide 39

smb.conf

You need this:

```
[global]
   workgroup = oscon
   server string = %h
   encrypt passwords = true
   security = user
   passdb backend = smbpasswd guest
   unix password sync = false
```

Slide 40

smb.conf

If you want *ichiban* to control the network:

```
# [global] continued
wins support = yes
os level = 144
domain master = yes
local master = yes
preferred master = yes
```

Slide 41

smb.conf

Recommended Linux options:

```
# [global] continued
socket options =
        IPTOS_LOWDELAY TCP_NODELAY
        SO_SNDBUF=4096 SO_RCVBUF=4096
```

Slide 42

smb.conf

Samba can talk to CUPS, in case your Windows machines can't:

```
   # [global] continued
   printing = cups
   printcap name = lpstat
   load printers = yes
```

Slide 43

smb.conf

The *magic* "printers" section.

```
[printers]
   comment = All Printers
   browseable = no
   path = /tmp
   printable = yes
   public = no
   writable = no
   create mode = 0700
```

Slide 44

smb.conf

The *magic* “homes” section.

```
[homes]
   comment = Home
   browseable = no
   read only = no
   create mode = 0775
```

Slide 45

smb.conf

Create a share called “commons”.

```
[commons]
   comment = The commons
   path = /home/commons
   browseable = yes
   read only = no
   create mode = 0775
```

Slide 46

4.5 Users and passwords

Samba needs a separate password database, and uses `smbpasswd` to change manipulate it.

I mentioned **usermin** before. Use the **changepass** module to allow users to change their passwords. It is the easiest way to keep Unix and Samba passwords in sync for all users.

Slide 47

5 Email

Your users should be able to email each other on the intranet.

For this, you could use a Mail Transport Agent on *ichiban*, and Mail User Agents (like MS Outlook) on the client machines.

Encourage users to use the common file area to transfer files, instead of attaching them to email!

Slide 48

Sendmail

Sendmail is probably still the core MTA on the Net.

Don't use it!

- Probably the most vulnerable software; even more than your Microsoft products!
- Difficult to configure.

Slide 49

Postfix

Postfix was designed to be secure and featurefull replacement for Sendmail. It's also much easier to configure.

But, all MTAs are not simple to setup, as there are so many different ways you might want to use them.

We'll try to keep it simple.

Slide 50

Postfix config files

The config files we might want to use are:

- /etc/postfix/main.cf
- /etc/aliases

There are many others. Again, we won't have time to cover them.

Slide 51

main.cf

```
mydomain = oscon
myorigin = /etc/mailname  # Debian
mydestination = $myhostname, localhost
append_dot_mydomain = yes
relayhost =  # a smarter MTA
home_mailbox = Maildir/
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
recipient_delimiter = +
```

Slide 52

aliases

```
root: marty+root
postmaster: marty+postmaster
webmaster: marty+webmaster
```

Run `newaliases`

Slide 53

5.1 SPAM

Another topic we could spend an entire day on, but can't.

With Postfix, there are 3 places we can try to detect and stop SPAM:

- at the SMTP receiver
- at the `content_filter`
- in each user's .forward file

Slide 54

5.2 POP and IMAP

There are too many different POP3 and IMAP servers! There are only a few that provide both services.

We'll use **dovecot** because it is fast, flexible, and almost simple.

Courier was a close runner-up. It's not as simple.

Slide 55

dovecot.conf

```
protocols = imap pop3
login = imap
login = pop3
default_mail_env = maildir:~/Maildir
maildir_copy_with_hardlinks = yes
auth = default
auth_mechanisms = plain
auth_userdb = passwd
auth_passdb = pam
auth_user = root
```

Slide 56

5.3 Receiving email

If *ichiban* is visible to the outside world, let the world know: setup an MX record for your domain (this won’t work for “oscon”).

If not, you might need to investigate **fetchmail**.

Slide 57

6 Conclusion

You now have a cheap and reliable network server.

Put *ichiban* in a corner, and forget about it.

I like GNU/Linux servers.