

"Pocket" ISP based on RedHat Linux

Table of Contents

<u>"Pocket" ISP based on RedHat Linux</u>	1
<u>Anton Chuvakin, anton@chuvakin.org</u>	1
<u>1.Introduction</u>	1
<u>2.Changes</u>	1
<u>3.TODO Mon Aug 7 03:39:28 EDT 2000</u>	1
<u>4.Step by step guide</u>	1
<u>5.Conclusion</u>	2
<u>6.References</u>	2
<u>1.Introduction</u>	2
<u>2.Changes</u>	3
<u>3.TODO Mon Aug 7 03:39:28 EDT 2000</u>	3
<u>3.1 New versions of this document</u>	4
<u>3.2 Feedback</u>	4
<u>3.3 Standard disclaimer</u>	4
<u>3.4 Support</u>	4
<u>3.5 Copyright information</u>	5
<u>4.Step by step guide</u>	5
<u>4.1 Get RH</u>	5
<u>4.2 Install RH</u>	6
<u>4.3 Some install tips</u>	6
<u>4.4 Some preliminary security configuration</u>	6
<u>4.5 Remove unnecessary services</u>	7
<u>4.6 Enable multiple IP addresses</u>	9
<u>4.7 Configure DNS</u>	10
<u>4.8 Configure httpd</u>	14
<u>4.9 Configure sendmail</u>	15
<u>Setup</u>	15
<u>Troubleshooting</u>	16
<u>4.10 Configure POP3</u>	16
<u>Setup</u>	16
<u>Troubleshooting</u>	18
<u>4.11 Configure FTP server</u>	18
<u>Anonymous FTP setup</u>	18
<u>Guest FTP setup</u>	20
<u>4.12 Configure dialin</u>	23
<u>Linux setup</u>	23
<u>Windows setup</u>	26
<u>4.13 Open access</u>	26
<u>5.Conclusion</u>	27
<u>6.References</u>	27

"Pocket" ISP based on RedHat Linux

Anton Chuvakin, anton@chuvakin.org

v1.1.0 1 August 2000

This document outlines the setup of a single RedHat box for dialins, virtual web hosting, virtual email, POP3 and ftp servers. Why anybody might need this in one box is beyond the scope of this document. The idea is a complete ISP solution based on RedHat Linux. Any part of this setup can be implemented separately though. I will try to emphasize all the commands so one can just paste them to configure his own box. The list of documents that I borrowed from and some further reading is provided below (see References section). I will try to keep security in mind on all stages of the setup and to make clear all the security limitations of this setup. I should add that assets that are to be protected in this case are considered not very valuable (e.g. personal pages etc) thus efforts spent on securing the setup are allowed to be limited.

1. [Introduction](#)

2. [Changes](#)

3. [TODO Mon Aug 7 03:39:28 EDT 2000](#)

- [3.1 New versions of this document](#)
- [3.2 Feedback](#)
- [3.3 Standard disclaimer](#)
- [3.4 Support](#)
- [3.5 Copyright information](#)

4. [Step by step guide](#)

- [4.1 Get RH](#)
- [4.2 Install RH](#)
- [4.3 Some install tips](#)
- [4.4 Some preliminary security configuration](#)
- [4.5 Remove unnecessary services](#)
- [4.6 Enable multiple IP addresses](#)
- [4.7 Configure DNS](#)
- [4.8 Configure httpd](#)

- [4.9 Configure sendmail](#)
- [4.10 Configure POP3](#)
- [4.11 Configure FTP server](#)
- [4.12 Configure dialin](#)
- [4.13 Open access](#)

5.Conclusion

6.References

1.Introduction

The guide assumes some familiarity with Linux functionality and general Linux/UNIX setup procedure (although not very detailed). Fully functional brain is also required for some stages of the procedure. All setup would be done manually (without the use of [linuxconf](#), [Webmin](#) or other tools). Not that those are bad or that there is anything wrong with them. The reasons for that are: 1) it is comparatively hard to give step by step directions that produce predictable results as these tools pretend they are intelligent and "know better" (also known as "Windows syndrome") 2) layout of tools changes with time and is different in some distributions 3) manual setup gives better understanding of system works (not that it is always required though) 4) some tools allow only limited configuration of Linux system or do not keep up with updated features of services they try to configure.

Another solution seems to be very promising. It is [virtfs](#) developed by [Afra Ahmad](#). Its main part is a perl script so it does not suffer from being a "black box". It will automatically configure all virtual services in a highly customizable fashion.

It is based on taking advantage of the chroot environment. A separate and smaller filesystem is created for each virtual server, and when a service is requested, the main server will chroot to the desired virtual server.

This method may take up more disk space, but it is much more flexible, especially when dealing with the services. For example, it is possible to have two different email accounts bob@vdomain1.com and bob@vdomain2.com (as you are dealing with two different passwd files). It might be essential for a bigger hosting site.

For more information please visit the Virtfs page at <http://www.prongs.org/virtfs>.

While many improvements are possible to the setup described in this HOWTO they might be described in later editions of this document – I just outline one possible way (accidentally, the one I used). The writeup is aimed at RedHat Linux, but with trivial changes can be used on any modern Linux distribution. The resulting configuration loosely follows the setup of some particular machines built by the author.

2.Changes

from 1.0.3 to 1.1.0

- Description of virtfs added
- Qpopper and WUftpd bugs described
- Double connections issue mentioned as requested by one of the readers
- POP-only accounts described
- References added

from 1.0.2 to 1.0.3

- Some spelling errors corrected (thanks to Eugene Shishkin for that)

from 1.0.1 to 1.0.2

- *Some* errors corrected (spelling)
- Method to chroot non-anonymous ftp users ("guest" users; those with password and usernames, but with access only to their home directories; used for *easy* web updates)
- References section updated
- Troubleshooting subsections added to two sections
- Qpopper update

from 1.0.0 to 1.0.1

- *Many* errors corrected (both spelling and factual)
- References section updated
- Minor changes in wording and syntax to improve clarity
- More security info added to several sections
- Windows configuration for dialup added

Next update planned at: upon request or when new program versions are released

3.TODO Mon Aug 7 03:39:28 EDT 2000

- More on security of all the services we install (clear text password, DoS by overflowing partition in mail and ftp, http access configs etc), including maybe the very basic ipchains setup (ipchains as safer alternative to tcp wrappers)
- Add info on POP3 and ftp tunneling via ssh (just for fun)
- Add troubleshooting subsections to various sections
- Add SSL-enabled Apache install and basic configuration

- PDF version is available on all linuxdoc mirror sites
- Add news server setup

3.1 New versions of this document

New versions of this document can be found at

<http://www.chuvakin.org/ispdoc>

3.2 Feedback

All comments, error reports, additional information (very much appreciated!!!) and criticism of all sorts should be directed to: anton@chuvakin.org

<http://www.chuvakin.org/>

My PGP key is located at <http://www.chuvakin.org/pgpkey>

Please direct spelling error comments to your friendly local spell checker.

If you plan to ask for **help**, see support section first.

3.3 Standard disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. Additionally, this is an early version, with many possibilities for inaccuracies and errors.

One of many possible setups will be described. In the Linux world, there is usually a number of ways in which to accomplish things.

As far as I know, only programs that under certain terms may be used or evaluated for personal purposes will be described. Most of the programs will be available complete with source under GNU-like terms.

3.4 Support

This is ridiculous, right? Who may ask for support after seeing such a comprehensive doc ;-) ?

Anyhow, if you are curious about some particular aspect of this setup or some of my writing is unclear, just drop me an email and I *might* answer it (at least, be assured that I will read it).

Now, if you require a phone, hand-holding style support or **my** work on **your** system, I *might* be able to

provide it on certain terms (if I have time and your case seems interesting enough ;-))

3.5 Copyright information

This document is copyrighted (c) 2000 Anton Chuvakin and distributed under the following terms:

- Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.
- All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.
- If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, at

greg@sunsite.unc.edu

4. Step by step guide

Ingredients needed:

- RedHat Linux distribution (the instructions are exactly applicable to RedHat 6.1 or 6.0 (I think, 6.2, as well, since there are no major changes) and with some minor changes to 5.x versions)
- [compatible](#) hardware (also known as a PC), that includes network card and modem (at least one)
- 3–256 IP addresses (as the machine will give out some IP addresses for modem callers and use others for virtual hosting more than 1 is needed, the upper number is the maximum number of IP-based virtual hosts allowed without recompiling the stock RedHat kernel, lower is one real IP, one modem and one virtual IP).
- some sort of permanent network connection (using some modems for dialin while providing the Internet access via another modem is considered *weird* and not recommended)

Here follows the procedure:

4.1 Get RH

Purchase or otherwise procure the RedHat 6.1 (further referred as RH, latest version number is 6.1 at the time of writing) distribution and [compatible](#) hardware. One can get a full RH CDROM for about \$2.00 including shipping and handling at <http://www.cheapbytes.com>. This version will not contain such luxuries as secure

web server and extra software. For those you should turn to [RedHat website](#). Or probably buying the PC with Linux RH pre-installed is an option for some.

4.2 Install RH

Install the RH following the *instructions on the package* (might be added here later). CDROM install is very easy to perform. I suggest using text-mode setup, in my case their new graphical one failed miserably. When asked about the installation type (Server/Workstation/Custom) choose Server or Custom (if you know what you are doing)—you can always add software later. Some other important installation decisions are outlined further. For RH 6.0 and 6.1 you might be able to add packages to Workstation setup as well, but in RH 6.2 all the server services are disabled and significant amount of tweaking is required—so only Server or Custom is strongly recommended.

4.3 Some install tips

If your hardware really is [compatible](#) the installation process will detect and configure it correctly. Otherwise, refer to corresponding documentation for troubleshooting network card, modem, video card, etc problems (mostly HOWTOs and mini-HOWTOs, some are in References section below).

If your network card is detected properly you will be asked for an IP address of your machine, gateway address and network mask and the address of the DNS server (might be your own machine if you plan to set it up this way). Have all this info handy. Also you will be asked for a machine name and domain name. We will use a sample domain name **you.com** and the machine will be named **ns** (that gives us a fully qualified domain name (FQDN) **ns.you.com**). You should use whatever domain you registered (see Setting Up Your New Domain Mini-HOWTO, link in References section below) and intend to use as your primary domain (not a virtual). For the gateway address we will use a sample 111.222.333.111 address. Gateway is likely the router that connects your machine (or your LAN) to the outside world.

Enable **shadow** and **MD5** passwords for greater security. First of those makes the file that contains encrypted passwords readable only to `root` user and the second allows longer and harder to crack passwords. As it will be a standalone machine do not enable NIS/NFS.

After installation finishes and machine reboots you will see the login prompt. Enter login and password (for the root account) and start configuring you new Linux station.

4.4 Some preliminary security configuration

First (and fast), add a line: `ALL : ALL` to your `/etc/hosts.deny` file. That would (to some known extent) prevent other people from accessing your machine while you are doing the configuration. That will also prevent you from doing the same. For further configuration efforts (that can be done remotely, by the way) [secure shell](#) is recommended. Download the RPM package for RH from one of the many sites and install it (as root) using: `rpm -U ssh*rpm` or similar command (depends upon the version). You will have to get both client and server packages (if you want to ssh from this machines as well as to this machine). Upon installation all necessary post-installation commands (like server key generation) are run automatically by the RPM

package. You will have to start server manually using command `/etc/rc.d/init.d/sshd start`. Some early versions of ssh-1 and also all versions of ssh-1 compiled with RSAREF library contain a buffer-overflow bug. Use ssh-2 or the latest version of ssh-1 without RSAREF. If you do this you will have to allow access using ssh from some trusted machine (described later) in `/etc/hosts.allow` file.

If you want to be really rigorous in your configuration pursuits go to single user mode by giving the command **init 1**, in this case all work is to be done locally and you would not be able to test your network-related configuration as network is not available in this mode.

To further enhance your security **ipchains** software (that is usually part of your Linux distribution) can be used (for that refer to IPCHAINS HOWTO, link in References). It takes quite a bit more efforts to configure it than TCP wrappers, although some automated tools are available for that too.

4.5 Remove unnecessary services

Now let's deal with unnecessary services. Please note that my idea of "unnecessary" might not be 100% same as yours.

1. Services started from `/etc/inetd.conf`:

comment out all the lines, but those

```
ftp      stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd  -L  -l  -i  -a
telnet   stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Check this by using the command: **grep -v '#' /etc/inetd.conf**

If you will be using the secure shell (ssh), telnet is also not necessary and can be removed. Secure shell can either be started as a daemon on system startup or as a service from `/etc/inetd.conf`. Default configuration (used by the RPM package) is to start it as a daemon. Sshd can be compiled to refer to `/etc/hosts.allow` file for access control. In this case, while you will not have it in your `/etc/inetd.conf`, it will still use the settings from `/etc/hosts.allow` and `/etc/hosts.deny`. The advantages of this method is faster connection as the sshd will not have to regenerate server key every time somebody connects. On the other hand, if you start it from `/etc/inetd.conf` it will be more isolated from the outside world. More lines will be added to `/etc/inetd.conf` as necessary (POP3 is one of those).

2. Services started on system startup from `/etc/rc.d` directory:

Check what services are running by using: **ps ax**. You will get something similar to the sample output below:

```
PID TTY          STAT       TIME COMMAND
  1 ?            S          0:04 init
  2 ?            SW         0:30 [kflushd]
  3 ?            SW         0:32 [kupdate]
  4 ?            SW         0:00 [kpiod]
  5 ?            SW         0:03 [kswapd]
  6 ?            SW<        0:00 [mdrecoveryd]
```

"Pocket" ISP based on RedHat Linux

```
296 ?      SW      0:00 [apmd]
349 ?      S       0:00 syslogd -m 0
360 ?      S       0:00 klogd
376 ?      S       0:00 /usr/sbin/atd
392 ?      S       0:00 crond
412 ?      S       0:00 inetd
454 ttyS0   S       0:00 gpm -t ms
533 tty2    SW      0:00 [mingetty]
534 tty3    SW      0:00 [mingetty]
535 tty4    SW      0:00 [mingetty]
536 tty5    SW      0:00 [mingetty]
537 tty6    SW      0:00 [mingetty]
667 tty1    SW      0:00 [mingetty]
4540 ?     S       0:00 httpd
5176 ?     S       0:00 httpd
5177 ?     S       0:00 httpd
5178 ?     S       0:00 httpd
5179 ?     S       0:00 httpd
5180 ?     S       0:00 httpd
5181 ?     S       0:00 httpd
5182 ?     S       0:00 httpd
5183 ?     S       0:00 httpd
7321 ?     S       0:00 /usr/sbin/sshd          <<< only after you install
7323 pts/0   S       0:00 -bash
7336 pts/0   R       0:00 ps ax
```

Lets concentrate on processes that listen to network, such as lpd. Since we do not plan to use our server for printing (we sure might, I just don't describe it here), I suggest we remove the printer daemon by: **rpm -e lpd** . If rpm complains about any dependencies (like, in my case, printfilter and rhprinttool), add them to your **rpm -e** command and repeat it. Other services that should be removed are NFS, NIS, samba etc, if they got installed by mistake. Again, these are useful things, I am just following the *golden rule* "**remove the software you don't currently use**". And, with RH RPM it is really easy to add it any time in the future.

Some more basic security settings can be obtained from [Armoring Linux](#) paper. As suggested there, lets make a wheel group with trusted users (in our case, only user youwill be able to do /bin/su and to run cron jobs (together with root).

- wheel group for sensitive commands:

1. vi /etc/group, add a line (if it doesn't exist):

```
wheel:x:10:root,you
```

If line exists, just add you in the end as shown. You don't have to use vi (and I understand it very well), just use your favorite editor (for a nice reasonably user-friendly non-X editor try pico, distributed together with mail program pine, the latter is part of most Linux distributions)

2. /bin/chgrp wheel /bin/su
change group ownership to wheel group on /bin/su
3. /bin/chmod 4750 /bin/su
change mode on /bin/su

- restrict cron:

"Pocket" ISP based on RedHat Linux

To only allow `root` and `you` to submit cron jobs create a file called `/etc/cron.allow` that contains usernames that you want to be able to run cron jobs. This file might look like this:

```
root
you
```

Why should one restrict cron jobs? Local exploits to elevate privileges to `root` from, say, `nobody`, exist for some versions of cron.

I suggest you do not install X Windows as it will bring new concern that you might not be prepared to deal with.

4.6 Enable multiple IP addresses

Now we are ready to enable our machine to handle multiple IP addresses for virtual hosting. At that point, the IP Aliasing HOWTO might come handy (see link in References). For several reasons, IP-based virtual hosting is better (if you have enough IP addresses, that is). For instance, reverse lookups would succeed, if done from the browser side. It might also be needed for hosting cryptographically enabled websites (commonly known as "secure websites"). Older browsers (not supporting HTTP 1.1) will get unhappy too.

The changes would be concentrated in `/etc/rc.d/` directory. To enable multiple IP addresses your kernel should support this. On a freshly installed RH Linux it does. To verify it one should look into the config file that was used to compile the kernel. In my case, it was `/usr/src/linux/configs/kernel-2.2.12-i686.config` since the machine has Pentium II processor. This file exists, if the `kernel-source` RPM package was installed. If line `CONFIG_IP_ALIAS=y` is present in the file than you are OK. While we are here, we can also confirm the ability to forward IP packets (needed for dialup users PPP). This ability is present, but not turned on by default. Also needed is the support for PPP protocol (line `CONFIG_PPP=m`, this means PPP support is compiled as a kernel loadable module, `CONFIG_PPP=y` is also OK)

The examples will use the ridiculous IP addresses 111.222.333.444–111.222.333.777 from C block 111.222.333.0. 111.222.333.444 is a real host IP (that is configured during RH installation), 111.222.333.555–777 are virtual addresses and 111.222.333.888 is a dialin user address (can be more of those).

Lets assume we want to configure 3 virtual hosts.

Two sets of commands will be used:

1.

```
/sbin/ifconfig eth0:0 111.222.333.555
/sbin/ifconfig eth0:1 111.222.333.666
/sbin/ifconfig eth0:2 111.222.333.777
```

These will bind the IP addresses to (virtual) interfaces `eth0:0`–`eth0:2`.

2.

"Pocket" ISP based on RedHat Linux

```
/sbin/route add -host 111.222.333.555 dev eth0
/sbin/route add -host 111.222.333.666 dev eth0
/sbin/route add -host 111.222.333.777 dev eth0
```

These commands will add routes for those addresses and connect those to real interface eth0 (ethernet card).

After doing them the ifconfig command output (ifconfig) will look like this:

```
eth0      Link encap:Ethernet  HWaddr 02:60:8C:4D:24:CE
          inet addr:111.222.333.444  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:901597 errors:33 dropped:0 overruns:0 frame:823
          TX packets:433589 errors:0 dropped:0 overruns:0 carrier:0
          collisions:128327 txqueuelen:100
          Interrupt:5 Base address:0x280

eth0:0    Link encap:Ethernet  HWaddr 02:60:8C:4D:24:CE
          inet addr:111.222.333.555  Bcast:111.222.333.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0x280

eth0:1    Link encap:Ethernet  HWaddr 02:60:8C:4D:24:CE
          inet addr:111.222.333.666  Bcast:111.222.333.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0x280

eth0:2    Link encap:Ethernet  HWaddr 02:60:8C:4D:24:CE
          inet addr:111.222.333.777  Bcast:111.222.333.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:5 Base address:0x280

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:26232 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26232 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

All commands can be added to the bottom of */etc/rc.d/rc.local* so that the changes are saved after reboot. Strictly speaking, rebooting machine is not required for adding new IP addresses. Please, do document all changes you do to your machines. Many a good sysadmin (or, should I say not-so-good?) were burned on that at some point in their careers.

4.7 Configure DNS

Now we are ready to configure DNS. The easiest way would be to add the hostnames (real and all the virtual) that we want to be seen by the world to the configuration of some machine that already has bind (DNS daemon) running. But, since we are setting up ISP-in-a-box we might not be able to avoid "DNS fun".

Now, let me also try to defend the (well, questionable) choice of "outdated" version of bind 4.9.7 (last of the pre-8 series). I know that my arguments can be beaten, but I consider bind 4.9.7 much more time-tested and stable. The arguments for upgrading to 8.x (provided <http://www.acmebw.com/askmrdns/00444.htm> and

<http://www.dns.net/dnsrd/servers.html> and, I guess, at many other places) still didn't seem to convince many people. And, lets not forget the "exploit of 1999" – ADMROCKS, that gives remote root access to almost any Linux machine running bind prior to 8.1.2 patch 3. Judging by the INCIDENTS mailing list, this is still a very popular way to attack RH versions 5.0–6.1 if no recommended upgrades are installed.

Here are the instructions, loosely following the DNS book from O'Reilly (a good one, highly recommended to all, but very casual DNS user).

1. Find and install bind 4.9.7 either from RPM package (RH 4.2, if I am not mistaken – for that you can use RPMFIND.net, personally I didn't try this and so I am somewhat skeptical about installing RH 4.2 package on RH 6.1 system, but it might work) or from source ([bind 4.9.7](http://bind.4.9.7), compiling it is a bit troublesome, but reading all the README files in the archive will definitely help).
2. Create files and directories needed for bind:

- ◆ */etc/named.boot*
- ◆ */etc/namedb*
- ◆ */etc/namedb/db.you*
- ◆ */etc/namedb/db.111.222.333*
- ◆ */etc/namedb/db.127.0.0*
- ◆ */etc/namedb/db.yoursite1*
- ◆ */etc/namedb/db.yoursite2*
- ◆ */etc/namedb/db.yoursite3*

This will be used for 3 virtual domains: **yoursite1.com**, **yoursite2.com** and **yoursite3.com**. One more important comment refers to secondary DNS issue. As all your domains and all their services will be hosted on the same machine, DNS backup in the form of secondary server doesn't make much sense: if your primary DNS is down everything else (mail, www, ftp, pop, etc) is down as well. But you do have to have a secondary DNS to register a domain. Try to convince somebody to put you in as a secondary or use a free DNS service (link is in Setting Up Your New Domain Mini-HOWTO).

3. That is how they look like (if you are unfamiliar with bind 4.x configuration file format, please, do read either the O'Reilly DNS book or any of the HOWTOs or documents at bind pages, or, better, all of the above. You also have an option of using them without understanding, but this is a bad idea in general):

/etc/named.boot

This is the main config file for bind 4.9.x.

```
directory /etc/namedb

;cache-obtained from internic, usually
cache . db.cache

;main config files
primary you.com db.you
;reverse lookups
primary 333.222.111.in-addr.arpa db.111.222.333
;localhost.localnet configs
primary 0.0.127.in-addr.arpa db.127.0.0
```

"Pocket" ISP based on RedHat Linux

```
;virtual Domains
primary yoursite1.net          db.yoursite1
primary yoursite2.net          db.yoursite2
primary yoursite3.net          db.yoursite3
```

4.

/etc/namedb/db.you

```
; defines our local hosts at you.com, just one in our case, and its aliases
@                IN          SOA      ns.you.com. root.ns.you.com. (
                2000012190 7200 1800 3600000 7200 )

;name servers and mail servers
                IN          NS       ns.you.com.
                IN          MX       10 ns.you.com.
                IN          A        111.222.333.444
ns              IN          A        111.222.333.444

;address of the canonical names
localhost       IN          A        127.0.0.1
gateway         IN          A        111.222.333.111

;aliases (to use in ftp: ftp ftp.you.com etc, for clarity)
www             CNAME      ns
mail           CNAME      ns
ftp            CNAME      ns
pop3           CNAME      ns
```

5.

/etc/namedb/db.111.222.333

```
;reverse mapping of our IP addresses
.
;origin is 333.222.111.in-addr.arpa
333.222.111.in-addr.arpa. IN      SOA      ns.you.com. root.ns.you.com. (
                1999121501 7200 1800 3600000 7200 )

;name Servers
                IN          NS       ns.you.com.

;addresses point to canonical name
444.333.222.111.in-addr.arpa. IN    PTR      ns.you.com.
;dialins
888                IN          PTR      dialup.you.com.

;virtual hosts
555                IN          PTR      yoursite1.com.
666                IN          PTR      yoursite2.com.
777                IN          PTR      yoursite3.com.
```

6. */etc/namedb/db.127.0.0*

```
;local loop config file
0.0.127.in-addr.arpa. IN      SOA      ns.you.com. root.ns.you.com. (
                1997072200 7200 1800 3600000 7200 )
```

"Pocket" ISP based on RedHat Linux

```
1                IN      NS      ns.you.com.  
                IN      PTR     localhost.
```

7. /etc/namedb/db.yoursite1

```
; yoursite1.com  
@                IN      SOA     virtual root.virtual (  
                1999092201    ; Serial: update each time t  
                7200         ; refresh, sec  
                1800         ; retry, sec  
                3600000      ; expire, sec  
                7200        ) ; minimum TTL  
  
;name servers  
                IN      NS      ns.you.com.  
                IN      MX      10 virtual  
                IN      A      111.222.333.555  
  
;address of the canonical names  
localhost       IN      A      127.0.0.1  
gateway         IN      A      111.222.333.111  
virtual         IN      A      111.222.333.555  
                IN      MX      10 virtual  
  
;aliases  
www             CNAME   virtual  
mail            CNAME   virtual  
ftp             CNAME   virtual  
pop3           CNAME   virtual
```

8. /etc/namedb/db.yoursite2

```
; yoursite2.com  
@                IN      SOA     virtual root.virtual (  
                1999092201    ; Serial: update each time t  
                7200         ; refresh, sec  
                1800         ; retry, sec  
                3600000      ; expire, sec  
                7200        ) ; minimum TTL  
  
;name servers  
                IN      NS      ns.you.com.  
                IN      MX      10 virtual  
                IN      A      111.222.333.666  
  
;address of the canonical names  
localhost       IN      A      127.0.0.1  
gateway         IN      A      111.222.333.111  
virtual         IN      A      111.222.333.666  
                IN      MX      10 virtual  
  
;aliases  
www             CNAME   virtual  
mail            CNAME   virtual  
ftp             CNAME   virtual  
pop3           CNAME   virtual
```

9. /etc/namedb/db.yoursite3

"Pocket" ISP based on RedHat Linux

```
; yoursite3.com
@                IN      SOA      virtual root.virtual (
                1999092201    ; Serial: update each time t
                7200          ; refresh, sec
                1800          ; retry, sec
                3600000       ; expire, sec
                7200 )        ; minimum TTL

;name servers
                IN      NS      ns.you.com.
                IN      MX      10 virtual
                IN      A      111.222.333.777

;address of the canonical names
localhost      IN      A      127.0.0.1
gateway       IN      A      111.222.333.111
virtual       IN      A      111.222.333.777
                IN      MX      10 virtual

;aliases
www           CNAME   virtual
mail         CNAME   virtual
ftp          CNAME   virtual
pop3         CNAME   virtual
```

These configuration files will allow you to host these three virtual domains and your real domain **you.com**.

4.8 Configure httpd

To server html pages httpd daemon is used. RH 6.1 comes with Apache 1.3.9 (latest version is currently 1.3.12). At that point it is wise to check RH site or its mirrors ([RH Mirrors](#)) for updates.

Most changes that we are about to make concentrate in */etc/httpd/httpd.conf* (RH standard location for Apache configuration). Default location for html pages (shown when you go to **www.you.com**) is */home/httpd/html*. You can allocate directories for virtual hosts within the same */home/httpd*, shown below are the following locations for them: */home/httpd/yoursite1*, */home/httpd/yoursite2* and */home/httpd/yoursite3*.

Below I provide the minimum necessary changes for your */etc/httpd/httpd.conf* file:

```
<VirtualHost 111.222.333.555>
ServerAdmin webmaster@you.com
DocumentRoot /home/httpd/yoursite1
ServerName www.yoursite1.com
ErrorLog yoursite1-error_log
TransferLog yoursite1-access_log
</VirtualHost>
```

```
<VirtualHost 111.222.333.666>
ServerAdmin webmaster@you.com
DocumentRoot /home/httpd/yoursite2
ServerName www.yoursite2.com
ErrorLog yoursite2-error_log
TransferLog yoursite2-access_log
</VirtualHost>
```

"Pocket" ISP based on RedHat Linux

```
<VirtualHost 111.222.333.777>
ServerAdmin webmaster@you.com
DocumentRoot /home/httpd/yoursite3
ServerName www.yoursite3.com
ErrorLog yoursite3-error_log
TransferLog yoursite3-access_log
</VirtualHost>
```

That configuration will cause all logs to be stored in one directory (whatever is specified as such) for all sites. If that is not desired the **ErrorLog** and **TransferLog** directives can be changed to point to the proper location separately for each virtual host. The pages for the "real" **www.you.com** will be stored in default location */home/httpd/html*.

For more information, look at <http://www.apache.org>, Apache http server homepage. They have a lot of support pages, including those for virtual hosting setup (both IP-based and name-based [uses just 1 IP address]). Also useful is Linux WWW HOWTO (link in References section), section on virtual hosting.

4.9 Configure sendmail

Setup

Now we will deal with sendmail. Again, proposed are the minimum necessary changes to the stock RH */etc/sendmail.cf* and */etc/sendmail.cw*.

1. look for the lines that starts from `Dj$w.foo.com` and change it to point to your main ("real", not virtual) server name (**you.com**, so it will look like `Dj$w.you.com`).
2. locate file */etc/sendmail.cw* and make it look like this

```
# sendmail.cw - include all aliases for your machine here.
you.com
ns.you.com
mail.you.com
yoursite1.com
mail.yoursite1.com
yoursite2.com
mail.yoursite2.com
yoursite3.com
mail.yoursite3.com
```

These are necessary so that sendmail accepts mail for these domains.

This **does not** address the issue of `user@yoursite1.com` and `user@yoursite2.com` mail getting to different mailboxes. For that look into */etc/mail/virtusertable* functionality (appropriate line in */etc/sendmail.cw* is `Kvirtuser hash -o /etc/mail/virtusertable`, detailed info may be added here later). Excellent documentation on that is on <http://www.sendmail.org/virtual>, sendmail reference on virtual hosting.

It is worthwhile to add that linuxconf proposes a somewhat different scheme for virtual email with separate spool directories for all domains (that cleanly solves the above "name-conflict" issue"), but that requires a special virtual-aware POP/IMAP server (included with RH) and is somewhat more complicated. It is recommended for bigger email volume sites with many users within each domain.

A few words about sendmail, it is a good idea (good from the security standpoint) to have sendmail run from *inetd.conf* and not as a standalone daemon. For that we need to add it to */etc/inetd.conf*, remove it from */etc/rc.d/init.d*, add the sendmail queue processing to cron. Here is what you have to do:

1. Add the following line to */etc/inetd.conf*:

```
smtp stream tcp      nowait root    /usr/sbin/tcpd  /usr/sbin/sendmail -bs
```

2. Edit */etc/rc.d/init.d/sendmail* to have `exit 0` somewhere in the very beginning (might not be the best way, be sure to document the changes you do to these files) so that this file does nothing instead of starting sendmail
3. By editing your (root's) crontab (to edit do **crontab -e**) add a line like this

```
* /20 * * * * /usr/sbin/sendmail -q
```

That would process sendmail queue every 20 min (if it exists). The described steps will simplify sendmail access control and will let you regulate who can talk to your 25 port, not just who can send email through you. The lines in */etc/hosts.allow* that let all machines from .com and .org domains send you email are as follows

```
sendmail: .com .org
```

Please, note, that the daemon name, not protocol name is used here (sendmail, NOT smtp).

That would allow your system to handle email for all those domains.

Troubleshooting

PROBLEM: mail that you are trying to send is denied with a message `Relaying denied`

SOLUTION: Look into your */etc/sendmail.cf*. Are you sure all possible variations of your hostname and of your virtual hostnames are here? Look in the message headers and see from what machine it was rejected from: does it look like another name of yours that you missed?

4.10 Configure POP3

Setup

POP3 configuration is easy (no "virtualization" is required for this setup). RH comes equipped with `imapd` IMAP server. If you do not want to use IMAP functionality or do not like this particular implementation (buffer overflow bugs were discovered in it at some point) the good idea is to use `qpopper`, free POP3 daemon from Eudora <http://www.eudora.com/freeware/qpop.html>. At the time of writing they release the "old" version (`qpopper 2.53`) and "new" (`qpopper 3.0`). It is important to note that versions earlier than 2.5 contain a buffer overflow error that allows remote root exploit to be executed. Same problem plagues "public betas" up to 3.0 release 21. Use either 2.53 or the latest 3.0 (the former is better audited and the latter is better

"Pocket" ISP based on RedHat Linux

suited for RH – seamlessly works with PAM authentication). I suggest using 3.0, so the instructions below apply to that case. As of April 13, Qpopper 3.0 is no longer beta, but a regular software. As of recently, the bug was discovered in Qpopper 2.53 that allows the attacker to obtain a shell with group-id 'mail', potentially allowing read/write access to all mail.

1. wget

```
ftp://ftp.qualcomm.com/eudora/servers/unix/popper/qpopper3.0.tar.Z
```

Retrieve the archive from Eudora site.

2. tar zxvf qpopper3.0.tar.Z

Uncompress and untar the contents.

3. cd qpopper

If you need explanation for this step, please, discontinue reading the document.

4. ./configure --enable-specialauth --with-pam --enable-log-login
--enable-shy

The options here are:

--enable-specialauth: allows MD5 and shadow passwords

--with-pam: allows the use of RH Pluggable Authentication Modules (PAM) technology

--enable-log-login: log successful logins, not only failures (not really that useful as it will use tcpd wrappers logging anyway)

--enable-shy: conceal version number (yeah, a little pesky manifestation of "security through obscurity")

5. make

That compiles the popper

6. /bin/cp popper/popper /usr/local/bin

Copies the binary to */usr/local/bin*

7. Now set the mode to

```
-rwx----- 1 root root 297008 Feb 16 15:41 /usr/local/bin/popper
```

by using the command:

```
chmod 700 /usr/local/bin/popper
```

8. Add a line to */etc/inetd.conf*

```
pop3 stream tcp nowait root /usr/sbin/tcpd /usr/local/bin/popper -s
```

That would cause the tcpd wrapper to control access to popper. The lines to add in */etc/hosts.allow* are

```
popper: .good.com .nice.org
```

That will allow people from domains *good.com* and *nice.org* to read email via POP3 client from your machine.

"Pocket" ISP based on RedHat Linux

To cause qpopper to use PAM authentication one must create a file for POP3 service in `/etc/pam.d/` directory. File should be named "pop3" (same as line in `/etc/services` and qpopper compile-time option). The file looks like this:

```
auth        required    /lib/security/pam_pwdb.so shadow
account     required    /lib/security/pam_pwdb.so
password    required    /lib/security/pam_cracklib.so
password    required    /lib/security/pam_pwdb.so nullok use_authtok md5 shadow
session     required    /lib/security/pam_pwdb.so
```

9. For whatever reason stock RH lists line in `/etc/services` file for POP3 protocol as "pop-3". And since qpopper prefers to see "pop3", it should be edited to be:

```
pop3        110/tcp        # pop3 service
```

That would allow all user to get their email via any reasonable mail client.

Troubleshooting

PROBLEM: you are connecting to your POP server with valid password and username and they are rejected with a message `Password incorrect`.

SOLUTION: PAM doesn't like your setup. This message is common for qpopper 2.53, use 3.0 and it should disappear. Otherwise, look into `/etc/pam.d/pop3` that you created. Is it OK?

4.11 Configure FTP server

Anonymous FTP setup

We will use only anonymous ftp and will not allow any non-anonymous user any access. Here we describe the anonymous ftp server setup that allows anonymous uploads. Any self-respecting guide on the subject will tell you that "this is a bad thing". But how is it worse than allowing users to ftp from untrusted location and transfer their passwords in clear text? Not everybody (especially, using Windows) can easily setup an ftp tunnel via ssh.

I suggest using the stock RH `wu-ftpd` (version 2.6.0 at the time of writing). While it is rumored that there are "more secure" ftp daemons (Pro-FTP?), `wu-ftpd` appears to be one most commonly used. Recently a series of bug was again discovered in `wu-ftpd` and its reputation as the most popular ftp daemon seem to be dwindling. CERT has issued an advisory concerning WU-FTPD and all ftp daemons derived from BSD's final release.

RH installs the `wu-ftpd` (package `wu-ftpd-2.6.0-1`) by default in server configuration. You are encouraged to check for updates as running ftp is an important security concern. There is also a separate rpm package that creates a separate directory for anonymous ftp home (`anonftp-2.8-1`). As anonymous ftp always does a `chroot()` system call (puts the user in the restricted file system) all necessary binaries and libraries are

"Pocket" ISP based on RedHat Linux

required. The typical directory looks like this (output of `ls -lRa` in `/home/ftp`):

```
.:
total 20
d--x--x--x  2 root    root    4096 Feb 15 06:22 bin
d--x--x--x  2 root    root    4096 Feb 15 06:22 etc
drwxrws-wt  2 root    wheel   4096 Feb 18 19:51 incoming
drwxr-xr-x  2 root    root    4096 Feb 15 06:22 lib
drwxr-sr-x  3 root    ftp     4096 Feb 15 23:34 pub

bin:
total 344
---x--x--x  1 root    root    15204 Mar 21 1999 compress
---x--x--x  1 root    root    52388 Mar 21 1999 cpio
---x--x--x  1 root    root    50384 Mar 21 1999 gzip
---x--x--x  1 root    root    29308 Mar 21 1999 ls
-----  1 root    root    62660 Mar 21 1999 sh
---x--x--x  1 root    root    110668 Mar 21 1999 tar
lrwxrwxrwx  1 root    root         4 Feb 15 06:22 zcat -> gzip

etc:
total 40
-r--r--r--  1 root    root     53 Mar 21 1999 group
-rw-r--r--  1 root    root   31940 Mar 21 1999 ld.so.cache
-r--r--r--  1 root    root     79 Mar 21 1999 passwd

incoming:
total 0

lib:
total 1212
-rwxr-xr-x  1 root    root    77968 Mar 21 1999 ld-2.1.1.so
lrwxrwxrwx  1 root    root         11 Feb 15 06:22 ld-linux.so.2 -> ld-2.1.1.so
-rwxr-xr-x  1 root    root  1031004 Mar 21 1999 libc-2.1.1.so
lrwxrwxrwx  1 root    root         13 Feb 15 06:22 libc.so.6 -> libc-2.1.1.so
-rwxr-xr-x  1 root    root    77196 Mar 21 1999 libnsl-2.1.1.so
lrwxrwxrwx  1 root    root         15 Feb 15 06:22 libnsl.so.1 -> libnsl-2.1.1.so
-rwxr-xr-x  1 root    root    33596 Mar 21 1999 libnss_files-2.1.1.so
lrwxrwxrwx  1 root    root         21 Feb 15 06:22 libnss_files.so.2 -> libnss_files-2.1.1.so

pub:
total 0
```

Notice though, that for whatever reason, RH puts a copy of `/bin/sh` in `/home/ftp/bin`. I do not feel good about having it there, so it is `chmoded` to 0 by `chmod 0 sh` (can also be removed completely, but RPM might be slightly unhappy if you attempt to remove the package afterwards).

Permissions on `/home/ftp` directories and files should be carefully considered. In the above example, all of the system files are owned by root and are only readable (executable where necessary) by all. Files in `bin` are only executable (as is the directory itself to prevent listing of its contents).

The interesting part is permissions on `pub` and `incoming`.

"Pocket" ISP based on RedHat Linux

Below follows the configuration file for ftp daemon (*/etc/ftppaccess*). It is well commented to the degree of being self-explanatory:

```
#ideas from <htmlurl url="ftp://ftp.wu-ftp.org/pub/wu-ftp/upload.configuration.HOWTO" na
#only allow anonymous users-no other classes defined
class anonftp anonymous *

#number of users restriction with message shown when too many
limit remote 10 Any /toomany.msg

#prevent uploads everywhere (for now)
upload /home/ftp * no

#display the contents of some files upon login/cd
readme README* login
readme README* cwd=*
message /welcome.msg login
message .message cwd=*

#log all file transfers DISABLED
#log transfers anonymous

#prevent these file operations for anon users
delete no anonymous
overwrite no anonymous

#fast cd and aliasing for the same reason (not really necessary, but convenient)
alias inc: /incoming
cdpath /incoming
cdpath /pub
cdpath /

#what is allowed in paths
path-filter anonymous /etc/pathmsg ^[-A-Za-z0-9_\.]*$ ^\.\ ^-

#prevent the retrieval of some file
noretrieve .notar

#allow upload with NO subdirectory creation by anon users
upload /home/ftp /incoming yes root wheel 0400 nodirs

#allow upload with subdirectory creation by anon users DISABLED
#upload /home/ftp /incoming yes root wheel 0400 dirs

#prevent anon users to GET files from incoming (you might not like it, but it
#is a good idea-to prevent some people from using your ftp server to store
#their own stuff, pics, warez etc)
noretrieve /home/ftp/incoming
```

That would allow only anonymous users to do downloads and uploads in somewhat (!) controlled manner.

Guest FTP setup

Guest FTP users are those that have valid usernames and passwords (unlike anonymous), but do not have access to the whole directory structure (unlike real ones). So they are chrooted after authentication. Guest users can do uploads in this configuration.

"Pocket" ISP based on RedHat Linux

Easy **21**–step directions for that are provided below ;–)

Software used: `wu-ftpd-2.6.0-3`

Sample username will be created: **ftpguy**, user ID=505.

Her group will be: **lusers**, group ID=701.

If you want more users of the same sort, they should be the members of the same group. For that it might be good to change the directory structure somewhat so that all of them use the same *passwd* file and the same static *ls*. But, for better separation you can give each of them their own files.

1. `adduser ftpguy`

creates an entry in */etc/passwd*

2. `passwd ftpguy` change password to whatever

3. Edit file */etc/passwd*, last line (that contains our new user) should look like this

```
ftpguy:x:505:701::/home/ftpguy/./:/etc/ftponly  
yes, that is "slash"-"dot"-"slash" after his home directory.
```

4. Edit file */etc/shells*, add line, below

```
/etc/ftponly  
This file doesn't have to actually exist.
```

5. Edit file */etc/group*, add line, below

```
lusers:x:701:ftpguy
```

6. `cd /home`

7.

```
chown ftpguy.lusers ftpguy  
this directory is created by adduser command
```

8.

```
cd ftpguy; mkdir etc bin ; chown root.daemon etc bin  
this creates a directory tree for chroot
```

9.

```
chmod 111 etc bin  
this sets very conservative permissions on directories within the chrooted tree
```

10.

```
cp ~/static_ls /home/ftpguy/bin/ls  
obtaining static (not calling any libraries) version of /bin/ls: this directory (  
http://www.stanford.edu/group/its-ccs/security/binaries/linux/redhat/) contains static version of  
many RH 6.1-compatible utilities, including ls (local copy is  
http://www.chuvakin.org/ispdoc/ls.gz here, gunzip ls.gz to run)
```

11.

```
cd bin ; chown root.bin ls
```

12.

```
chmod 111 ls  
this sets very conservative permissions on binaries within chroot
```

13.

```
cd ../etc
```

14. Create file */home/ftpguy/etc/passwd* as follows

```
root:*:0:0:::/etc/ftponly
```

"Pocket" ISP based on RedHat Linux

```
ftpguy:*:505:701::/home/ftpguy/./:/etc/ftponly
```

15. Create file `/home/ftpguy/etc/group`, contents follow

```
root::0:root
lusers::701:ftpguy
```

16. `chown root.daemon passwd group`
this sets proper ownership of these files
17. `chmod 444 passwd group`
this sets minimum necessary permission on that file
18. `cd ~ftpguy; touch .forward`
this creates `.forward` file
19. `chown root.root .forward ; chmod 400 .forward`
and locks it for security reasons
20. `cd /etc`
21. Add the facilities for handling guest users into `/etc/ftpassess`

```
#-----
class anonftp guest,anonymous *

delete      no    anonymous,guest      # delete permission?
overwrite   no    anonymous,guest      # overwrite permission?
rename      no    anonymous,guest      # rename permission?
chmod       no    anonymous,guest      # chmod permission?
umask       no    anonymous,guest      # umask permission?

guestgroup lusers

limit remote 10 Any /toomany.msg
upload /home/ftp * no
readme README* login
readme README* cwd=*
message /welcome.msg login
message .message cwd=*

alias inc: /incoming
cdpath /incoming
cdpath /pub
cdpath /

path-filter anonymous /etc/pathmsg ^[-A-Za-z0-9_\.]*$ ^\ ^-
noretrieve .notar
upload /home/ftp /incoming yes root wheel 0400 nodirs
noretrieve /home/ftp/incoming
```

Lets test this beast:

```
localhost[anton]#1008: ftp localhost
```

"Pocket" ISP based on RedHat Linux

```
Connected to anton.
220 anton FTP server (Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000) ready.
Name (localhost:anton): ftpguy
331 Password required for ftpguy.
Password:
230 User ftpguy logged in. Access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 4
drwx-----  4 505      701          1024 Apr  8 02:16 .
drwx-----  4 505      701          1024 Apr  8 02:16 ..
-r-----  1 0         0              0 Apr  8 02:16 .forward
d--x--x--x  2 0         2          1024 Apr  8 02:09 bin
d--x--x--x  2 0         2          1024 Apr  8 02:15 etc
226 Transfer complete.
ftp> mkdir TEST
257 "/TEST" new directory created.
ftp> ls -l
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 3
-r-----  1 0         0              0 Apr  8 02:16 .forward
drwxr-xr-x  2 505      701          1024 Apr  8 02:32 TEST
d--x--x--x  2 0         2          1024 Apr  8 02:09 bin
d--x--x--x  2 0         2          1024 Apr  8 02:15 etc
226 Transfer complete.
ftp>
```

and so on.

4.12 Configure dialin

Now the fun part starts. We want the machine to allow dial-in access via attached (inserted?) modem or modems. It will provide either regular shell or restricted shell (that only executes pppd daemon). Windows 95/98 users should be able to effortlessly dial in using all default settings of their computers.

Linux setup

To handle login via serial line some version of `getty` program is needed. This program monitors the serial line (`/dev/ttyS1` will be used throughout the document, see serial HOWTO for details) and upon connection shown the login prompt or starts a program.

I suggest using the `mgetty` program (as it has more features and is easier to setup than some of the competitors).

RH comes with `mgetty-1.1.21-2`, that also has extensions to receive faxes and voice mail (if the modem supports this). Check whether `mgetty` is installed by doing: `rpm -qa | grep mgetty`.

"Pocket" ISP based on RedHat Linux

After installing mgetty some reconfiguration is necessary. The files that should be changed and the details follow:

1. */etc/inittab*

That enables mgetty to start when system is booted and be respawned accordingly. These lines should be added in the end.

```
#for dialins use mgetty
#note this S1 in the beginning of the line and ttyS1 in the end
S1:2345:respawn:/sbin/mgetty ttyS1
```

2. */etc/ppp/options*

This file controls the pppd daemon whenever it is started. Some of the options here are optional (hey, that why they are called options, right?).

```
auth -chap +pap login modem crtscts debug proxyarp lock
ms-dns 111.222.333.444
```

Here is their brief meaning:

- ◆ **auth** : use some sort of authentication for dialin clients
- ◆ **-chap**: not CHAP
- ◆ **+pap**: use PAP
- ◆ **login** : use the system password file for authenticating the client using PAP and record the user in the system wtmp file, */etc/ppp/pap-secrets* should still be present (see below)
- ◆ **modem** : use the modem control lines (for carrier detection and other stuff)
- ◆ **crtscts** : use hardware flow control
- ◆ **debug** : log extra info (might be removed after everything is fine)
- ◆ **proxyarp** : this is needed to connect from the client to the Internet, not just to the LAN you dialed into
- ◆ **lock**: pppd should create a lock file for the serial device
- ◆ **ms-dns 111.222.333.444**: this info is provided to Windows box as a default DNS server

Look at pppd man page for all the juicy details (parts of the above info is adapted from there)

3. */etc/ppp/options.ttyS1*

This file serves purpose similar to the previous one, but only applies to particular modem line. It specifies the IP address given to the remote machine (dynamic, in some sense, if you have more than one line) and the local IP as well.

"Pocket" ISP based on RedHat Linux

111.222.333.444:111.222.333.888

4. */etc/mgetty+sendfax/login.config*

This file is the main mgetty control file. Mgetty is Windows-PPP-aware, so it has provisions to start pppd automatically upon receiving connect from the Windows machine.

These lines should be present:

```
/AutoPPP/ - - /usr/sbin/pppd
```

Before adding them, check that some other version of similar command is absent there (commented out by default).

5. */etc/ppp/pap-secrets*

This is similar to */etc/password* file, but only used for dialins and contains **plain text passwords** (apparently, only visible to root). All users that you want to be able to dialin must have their usernames and password listed in this file. They should enter the same username and password into Windows Dial Up Networking configuration.

```
# Secrets for authentication using PAP
# these two users below can use dialin
# client      server  secret pword  remote IP addresses
dialinuser1   *      blabla!?!  111.222.333.888
dialinuser2   *      p8sSw0rD   111.222.333.888
```

Check that mgetty is running by looking for similar line in the output of `ps -ax` command.

```
4625 ?          S          0:00 /sbin/mgetty ttyS1
```

Now this machine will allow modem calls from any Windows 95/98 box.

As was noted by one of the readers some steps are to be taken to prevent users from sharing their dialin password with others. A simple perl/shell script will do the job by killing and logging connections that use the same username.

Also, if it is desirable to prevent users from using dialing in their usernames should not be put into */etc/ppp/pap-secrets*.

Windows setup

This is **really** straightforward.

1. Click on **My Computer**
2. Click on **Dial Up networking**
3. Click on **Make New Connection**
4. Proceed according to directions, enter the phone number etc
5. After a new connection is created click on it and enter the username and password (same as mentioned in */etc/passwd* and */etc/ppp/pap-secrets*)
6. Click **Connect** and it should work (it did in my case ;-)

4.13 Open access

Now, after testing all the services, we are ready to open the access to this machine. The main access control facility in our case is TCP wrappers (tcpd). Their behavior is controlled by 2 files */etc/hosts.allow* and */etc/hosts.deny*, as was mentioned in the sections devoted to various network services. TCP wrappers configuration can be done in 2 distinct ways and we will employ the simplest.

Let our */etc/hosts.deny* contain `ALL:ALL` clause, thus denying the access to all services (started from */etc/inetd.conf*) for all hosts and all users on them. Now we can allow what we need explicitly in */etc/hosts.allow*, thus following the philosophy "**what is not expressly allowed is denied**".

Lets assume we want to allow people to read and send email, we want some trusted hosts to update contents of the web pages and we want admin workstation to have full access. So we arrive at the following */etc/hosts.allow*:

```
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
ALL: 127.0.0.1 adminbox.some.net
#we rely on anti-relaying features of sendmail 8.9.x to fight spam
#and also restrict some sites that we don't want to see email from
sendmail: ALL EXCEPT .kr
popper: .com .edu .gov .mil
#these people can upload/download stuff
in.ftpd: .this.net .that.net
```

5. Conclusion

There must be the conclusion, right?

6. References

Useful LDP HOWTOs (well, actually, all others are useful too)

1. [Setting Up Your New Domain Mini-HOWTO](#), really good guide of DNS setup and general network setup (recommended reading)
2. [Linux WWW HOWTO](#), provides more details on Apache setup, including virtual hosting
3. [Red Hat Linux 6.X as an Internet Gateway for a Home Network](#), some hints on network setup
4. [IP Aliasing On A Linux Machine](#), used for multiple IP on the same interface
5. [Ethernet HOWTO](#), look here in case of network card trouble
6. [IPCHAINS HOWTO](#), turn to this if more security is desired
7. [Serial HOWTO](#), serial ports, lines, modems and related stuff
8. [PPP HOWTO](#), some notes on PPP server setup

Software (used or mentioned) websites

1. [Eudora POP3 server](#)
2. [WU-FPTD ftp server](#)
3. [Sendmail MTA](#)
4. [Mgetty pages](#)
5. [Apache httpd server](#)

Other documents

1. [Armoring Linux](#)
2. [Setting Up POP/PPP server](#)
3. [Mgetty and Windows dialin info](#)
4. [Using RedHat 5.1 to Start an ISP](#), the short article on how to start an ISP if all you have is a Linux RH ;-)
5. [Guest FTP server setup](#)
6. [Linux Dialin Server Setup Guide](#) Yet Another Guide about that
7. [virtfs](#) a nice automatic tool for configuring virtual services based on Perl script
8. [Linux Public Access HOWTO](#) an old and not updated for 5 years document describing Linux-based ISP, some nice hints on equipment (serial boards) and performance

Resources, not related to the topic of the document ;-)

1. I also maintain a list of computer/network security related books with (where available) reviews and online availability. It is posted at <http://www.chuvakin.org/books>. If you have a book that I don't list please use the form on the page and I will add it to the list and maybe review it later.

