Building Your Own IP PBX

a Jupiterweb™ Networking eBook

Building Your Own IP PBX

If your business is like most businesses, your phone system is a vital resource. But, as with most businesses, chances are it's an area where you're always looking to save money.

With voice over IP (VoIP) popping up in so many segments of the consumer telephony market, you probably have some idea of the advantages it offers -- especially operating cost-savings.

Replacing a phone system is an intimidating prospect, conjuring up images of armies of technicians invading your premises and large bills arriving in the mail. It doesn't have to be that way. There are plenty of commercial vendors who'd love sell you their solutions, but there's also an alternative -- a free, open-source IP PBX product called Asterisk. (Why Asterisk? Because the asterisk, or star key, on your telephone keypad is the first character in phone commands.) Not only is Asterisk free, it works -- and there's a large community of resources that's grown up around the basic software package, making it safe and simple to make the move.

Is rolling your own PBX a far-fetched notion? Not at all. Thousands of businesses of all sizes are running Asterisk PBXs today, just as many businesses are running their company Web sites on the free, open-source Apache Web server.

Asterisk is licensed under the GPL. It is both a development toolkit, and a full-featured telephony server. Because Asterisk supports multiple protocols and integrates PSTN with VoIP, allowing you to mix and match analog, digital, and IP phones, you can migrate away from your existing PBX at a comfortable pace. Or, if you prefer, build a brand-new system, adding features and capacity at your own speed.

Asterisk gives you complete control of your telephony. You can run your Asterisk PBX yourself, or hire help, or purchase a commercial implementation. If you have the programming chops, you can even modify the source code to fix bugs or add new features.

Free, Not Stripped Down

Don't be put off by the free price tag. Asterisk is at least as sophisticated as most commercial PBXs, and often more so.

If all you want to do is replace your existing PBX and duplicate its functionality, Asterisk will do the job, and likely do it better and more easily. It also features voicemail; allows you to add/remove users; send voicemail to e-mail; conferencing; interactive voice response; call queueing; distinctive rings; user monitoring; and more.
Want free long-distance? Suppose you have a remote branch office that you're racking up big phone bills to talk to. Put an Asterisk server at each end and you can talk all you want. Strictly speaking, it's not free -- you need a broadband Internet connection to make it work, but if you already have one, or even better, have a nice dedicated high-speed WAN, it's an easy choice.

Want to build a sophisticated call center for cheap? You can build one with Asterisk for the cost of PC headsets, the Asterisk server, and other networking hardware.

Implementing Asterisk

It is unwise to rush out and start ripping out your existing PBX equipment. Telephony is complex, so you'll want to start slowly and take small steps.

For one thing, Asterisk runs on Linux, BSD, and MacOSX, so you'll need to be familiar with one of these operating systems.

Want to build a nice Asterisk test lab with a minimum of hassle? Get Asterisk@Home. Don't be misled by the name. Asterisk@Home is a complete Asterisk implementation with an excellent graphical management interface, so you can be up and running in less than an hour. A three-PC local test lab and an Asterisk installation at a remote location will let you test most of Asterisk's functions.

Let's put together a sample system for a 10-person office currently equipped with analog phone lines

Resources needed:
- A computer (Asterisk server)
- A broadband Internet connection
- An interface card to connect to the PSTN
- Adapters for your analog phones, or
- New IP phones
- A commercial VoIP service

Pricing the Basics
Your Asterisk software must run alone on a PC; the machine cannot be shared. (While Asterisk versions are available that run on Linux, the BSD Unixes, and Mac OS X, please note that driver support for the various interface cards is the strongest in Linux.)

For this scenario an ordinary middle-range PC works fine -- something with at least a 1.5 GHz CPU, 512 Mb of RAM, an Ethernet card, and at least a 20-Gb hard drive.

VoIP calls consume between 20 and 90 kbps each way. A typical business DSL service costs around $80/month for 1.5Mbps/896kbps (down/upstream). If your 10 users all jump on the phone at the same time, they could theoretically saturate your uplink: 10 x 90kbps = 900. But that's unlikely, so this type of DSL service should work fine.

To connect to your main phone line (analog trunk line), you'll need an adapter with an FXO port (FXO gateway) on the Asterisk server -- something like the Handy Tone 488. These cost around $80. The Handy Tone comes with a raft of excellent features; it's more than just a dumb interface.

You may keep your existing fleet of analog phones by using ATAs (Analog Telephone Adapters). These are also called FXS-to-Ethernet gateways, because they connect your analog phones to your computer network. One example is the Linksys SPA-1001, which costs about $60.

Beware of VoIP products that are linked to certain commercial services. For example, some Linksys devices work only with Vonage. Don't chain yourself to a single service provider.

You may choose to purchase new IP phones instead of ATAs. The prices on these vary, from around $70 for bare-bones phones to several hundred dollars for "PBX" phones. The sweet spot for value and quality is between $100 and $200; you have a lot of good choices in this range.

One of the big attractions of VoIP is the promise of free worldwide long distance. Call anywhere anytime over the Internet for nothing. What could be sweeter?

It's a pleasant dream, but the reality is that while you can escape the tyranny of long distance charges, it will still cost something. You have to pay for bandwidth and equipment, and invest some time and skill in running your Asterisk server. And the telcos are understandably unhappy at the idea of losing all that revenue, even as we still use their wires.

Suppose you have far-flung branch offices, or vendors or other business partners that you need to talk to a lot. You can set up your own private network of Asterisk servers and bypass the telcos entirely. In typical Asterisk fashion there are a number of ways to do this.

Free World Dialup (FWD) is a free central directory service that lets you easily find and connect to other VoIP users. You may connect either with an IP phone, or your Asterisk server. FWD supports both voice and video transmissions.

Connecting your Asterisk@Home server to use FWD is fairly simple. First you register for a FWD account then configure your server, and then you're ready to make and receive calls. Remember, this is a VoIP service only -- it does not give you access to the PSTN.

DUNDi (Distributed Universal Number Discovery) protocol is a peer-to-peer system for finding Internet gateways to telephony services. It operates like a blend of DNS and routing, only there is no central authority analogous to the root DNS servers. All participants publish their own authoritative routing information and share it with authorized peers. When Server A wants to know how to connect to Server B, it asks around until it receives an answer. Then it stores the information so that it can also respond to requests. You have complete control over what information and resources you choose to share.

Nearly any services that an Asterisk server provides can be made available to other peers. One way to test this and be part of an existing peer network is to join the DUNDi-test network, a free, open test network that includes PSTN termination. To prevent abuse, everyone who joins this network is required to sign and agree to abide by the General Peering Agreement, which you will find on Dundi.com. It contains instructions on how to execute it.

--Carla Schroder, VoIPPlanet.com

Finally, you need a commercial VoIP service provider, or someone who provides "PSTN service termination." This is necessary so you can call any phone number and not be limited to other VoIP users. Coverage and prices vary a lot, so shop around. Be sure to look for a provider that supports customer-owned equipment, aka "BYOD." Broadvoice charges BYOD customers $5.95/month.

Adding it up, our 10-person office will spend $1,100 to $2,500 on hardware, and have monthly expenses of maybe $86 for broadband and commercial VoIP services.

Bigger Systems

If you are fortunate to have a nice T1/T3 line, you'll get better service quality and more capacity. T1/T3 can be divided into separate voice and data channels, so routing and QoS are easy to manage. Your service provider should be your first stop. Find out what sort of voice/data services are offered, and what kind of deals they are willing to make to keep you happy, such as free interface hardware and bundle discounts.

Linux and the BSD Unixes have powerful routing engines and traffic shaping built-in, so you don't need separate routers. Of course, the more users you plan to support, the more powerful your Asterisk server hardware needs to be and the more storage you'll need. A computer with an Athlon 64 3000 CPU, 1 gigabyte of RAM, and a three-disk SATA RAID5 array with a hardware controller will run around $1,200, and ought to handle 50 or more medium-talkative users.

You'll need an interface card that supports both voice and data over your T1/T3, like the Digium Wildcard TE110P. This supports up to 50 users. The TE110P can be uplinked to another TE110P card, so you have an easy upgrade path as your user base grows. Digium is the sponsor of Asterisk, and provides an extensive line of both analog and digital telephony hardware.

FXO gateways (also known as PSTN interfaces) come in several sizes, from the single-port Handy Tone 488 to the four-port Audiocodes MP-104-FXO, for about $950. You need one port per analog trunk line.

Deciding what type of telephones you want to use, how robust your Asterisk server needs to be, how many Asterisk servers you need, and how much bandwidth you need depends on so many different factors it's hard to give simple answers. Please visit the Asterisk dimensioning page (http://www.voip-info.org/wiki/index.php?page=Asterisk+dimensioning) for a number of great real-world examples.

Building a Test Lab

Deploying a new Asterisk PBX is not a trivial task, so the wise admin first sets up a test lab. This can be completed in about an hour, and should cost little or no money. You should have knowledge of basic networking and Linux system administration.

It doesn't take much to set up a test lab. A minimal setup requires:

- For the Asterisk server: a PC with a Pentium III CPU or equivalent, a 10-gigabyte hard drive, a network interface card, and 256 megabytes of RAM . Do not share this machine; use it only to run Asterisk.
- Two client PCs equipped with network cards, softphones, soundcards, speakers, and microphones or headsets.
- A hub or switch to connect the three computers.

Our Asterisk installation will completely overwrite the hard drive, so back up anything you want to save.

Softphones are software VoIP clients, like the excellent SJPhone, which runs on Linux, Mac OS X, and Windows. USB headsets are nice, and you don't need a sound card if you use one of these. Of course, you may test any hardware you like, such as analog phone adapters, IP phones, and various types of server interfaces.

Getting the Software

We're going to be using Asterisk@Home. It's a sophisticated, customized Asterisk implementation that is perfect for the enterprise. It includes:

- Asterisk PBX
- Asterisk Management Panel, a Web-based graphical management interface. Asterisk contains several dozen configuration files, so AMP will save your time and sanity many times over
- Flash Operator Panel, a Flash-based, real-time monitor for watching and managing all PBX activity
- CentOS Linux CentOS is a free clone of Red Hat Enterprise Linux, so it's a stable, mature, heavy-duty server operating system
- OpenSSH for secure encryption
- SugarCRM for managing contacts. SugarCRM integrates phone calls, text messages, faxes, emails, and tasks and scheduling
- Festival Speech Engine, for rendering text-to-speech

You may download either an .iso image to create a bootable installation CD, or a compressed .tar archive to install on an existing Linux or Unix server. We'll use the .iso, since that is the fastest and easiest. It's about a 509-megabyte download. Get the most recent stable version; don't use the beta versions unless you know what you are doing.

Installing Asterisk@Home

Once you have created your installation CD, use it to boot up your Asterisk server. Remember, this overwrites your entire hard drive. First CentOS will install. The entire installation is automated -- you won't partition or select packages. You do need to be present when the CentOS installation is finished, because you'll need to remove the installation CD. After reboot, the Asterisk@Home installation will take place. It takes around 30 minutes.

Configuring the Asterisk Server

Your first chore after installation is to change the root password. Login to Asterisk using the default root login, which is the username "root" and the terribly secret password "password." Then use the passwd command to create a new password:

```
# passwd
Changing password for root
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Next, configure networking. If your Asterisk@Home machine is on a subnet served by a DHCP server, the installer will get its networking configuration from the DHCP server. If you don't have a DHCP server, networking will not be configured.

Either way you should give your Asterisk@Home server a static IP. Do this with the netconfig command. This brings up a graphical configuration menu. Make sure that "Use dynamic IP configuration (BOOTP/DHCP)" is not checked. Then enter your chosen IP address, netmask, default gateway, and primary nameserver. You should have Internet access, so the default gateway is the IP of your Internet gateway, and the primary nameserver is either the DNS server of your Internet provider, or a local caching nameserver.

When you're finished, restart networking to apply the changes:

```
# /etc/init.d/network restart
```


This is a good time to assign IPs to the client PCs so they are on the same subnet as the Asterisk server, and to connect all the computers to the hub or switch if you haven't already.

Now you want the Asterisk Management Portal. Fire up a Web browser on one of the client PCs and enter http://[asterisk IP address]. This opens the AMP Web management page. Click on "Asterisk Management Portal (AMP)" to log in. The default AMP user is "maint", and the default password is "password".

We're going to use the IP address 192.168.1.10 for the Asterisk@Home test server. You will need to substitute your own IP address.

Changing the AMP Password

Asterisk@Home comes with a handy script for changing the default AMP password, which is "password." Log into the server as root then run this command:

```
# passwd-maint
-----------------------------------------
Set password for AMP web GUI and maint GUI
User: maint
-----------------------------------------

New password:
Re-type new password:
Updating password for user maint
```

Starting and Stopping Asterisk@Home

To shutdown or reboot the server, fire up AMP and click the Maintenance command. You'll see the server status -- four green bars are what you want to see here -- and Reboot and Shutdown buttons.

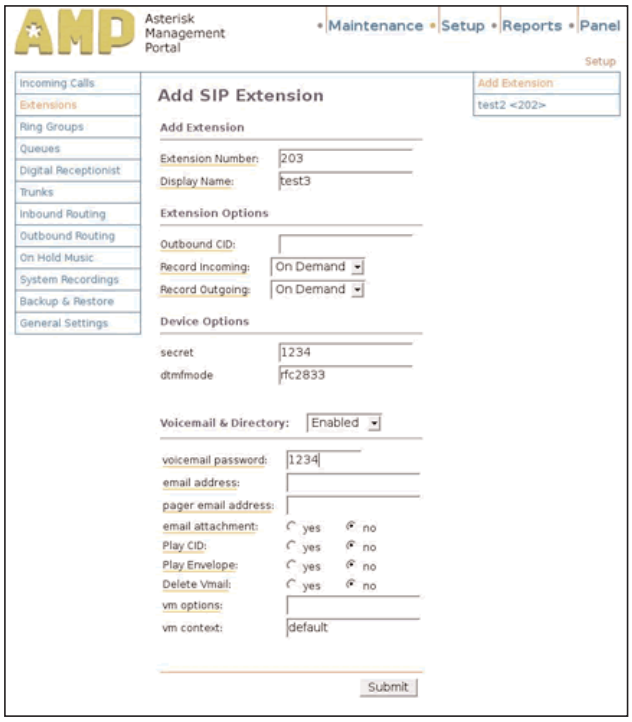
Local Asterisk Testing

To start out, get softphones and USB headsets for the clients. There are dozens of softphones with all sorts of feature sets and price ranges. Some only work with specific VoIP providers, so be careful what you get. We'll use the CounterPath X-Lite phone; it's free and runs on Linux, Mac OSX, and Windows. USB headsets are inexpensive and save a lot of hassles; they will obviate the need a sound card on the PC, and sound quality is decent.

First, we'll set up two new extensions on the Asterisk server. In AMP, click the Setup tab. Find the General Settings tab on the left-side menu. Hover the cursor over the different options to activate the tooltips.

Now let's set up two extensions for the two test clients. Click the Extensions button then select SIP. SIP (Session Initiation Procotol) is the most common VoIP protocol. Fill it out like the screen in Figure 1.

Figure 1



While you're testing, it might be easier to use the same password for both the login (which is entered in the "secret" box) and voicemail. The "secret" can be any standard combination of letters and numbers; for the voicemail password, be sure to use numbers only, since it will be entered on a telephone keypad.

When you're finished, click the Submit button. You'll see a red bar across the top of the screen that you must click to apply the changes. Add a second user in the same manner.

Now we'll configure the two clients.

Configuring the X-Lite Phone on Linux
Download and unpack the X-Lite softphone into whatever directory you want to run it from. It's a single executable. Start it up from the directory it is stored in with this command:

```
# ./xtensoftphone
```

When it runs for the first time, you'll see this:

```
$ ./xtensoftphone
I/O warning : failed to load external entity "/home/carla/.Xscrc"
```

No worries, ignore it. The phone will open, and a wizard will appear to walk you through sound testing and adjustment. Then it opens the screen where you enter your user settings. Using our example from Figure 1, enter this information:

```
Enable: yes
Username: 202 (your extension)
Authorization User: 202
Password: 1234 [your login password, or "secret"]
DomainRealm: 192.168.1.20 [your Asterisk server IP]
SIP Proxy: 192.168.1.20 [your Asterisk server IP again]
```

Figure 2

Now close out the configuration screen and the telephone. Then open the phone again with the ./xtensoftphone command. You should see something like Figure 2.

It logs in to the server as soon as you start it up. Now you can perform an echo test. Dial *43 and click the green phone icon. You will hear a woman's voice explaining how to perform the test. Just speak, and everything you say is echoed back to you. Click the red icon to hang up. Anytime you wish to change the settings, run ./xtensoftphone and click the little icon to the right of the Clear button. This opens the settings menu. Go to System Settings-> Sip Proxy.

Confusingly, you'll see other documentation that tells you that the echo test command is *45. This is incorrect, and you'll get a busy signal if you try it.

Configuring the X-Lite Phone on Windows and Mac OSX
The configuration screens are just the same as on Linux. The two main differences are you won't have an audio set-up wizard run the first time, and menu icons are created for you.

Testing Local Calling

Now you have a real live functioning local PBX. To call other extensions, dial the extension number. Leave messages and retrieve voicemail. To configure or fetch your voicemail, hit *98. You'll be prompted for your extension number and voicemail password.

Before we connect to the outside world, let's replace the stock Asterisk@Home logo with a logo of your own. You might want to do this just to put your company identity on your Asterisk server, or you may need to reassure a nervous boss who thinks that the name "Asterisk@Home" means it is not suitable for the enterprise.

Name your logo aaw_logo.png, then copy your logo to the /var/build_aah/www/ directory on the server. Asterisk@Home comes with an SSH server already running, so you can use this command to copy the file from a second PC on your LAN. Of course, you must use your own server IP or hostname:

```
$ scp aaw_logo.png root@aah_server1:/var/build_aah/www/
```

Now you must log in as root on the Asterisk server. You can do this from the LAN neighbor as well:

```
$ ssh root@aah_server1
root@192.168.1.20's password:
Last login: Tue Apr 11 17:52:43 2006 from 192.168.1.10
Welcome to Asterisk@Home
--------------------------------------------
For access to the Asterisk@Home web GUI use this URL
http://192.168.1.20
For help on Asterisk@Home commands you can use from this
command shell type help-aah.
[root@asterisk1 ~]#
```

Then download and execute the aah-change-logo script (http://www.voip-info.org/users/415/415/images/396/aah-change-logo.sh.txt), using these commands:

```
# wget http://www.voip-info.org/users/415/415/images/396/aah-change-logo.sh.txt
# dos2unix aah-change-logo.sh.txt
# sh aah-change-logo.sh.txt
```

The script finds and replaces all instances of the logo, so when you're finished you'll see your own logo in AMP. Figure 3 shows what it looks like with an "Asterisk@Work" logo.

Figure 3

Making Internet Phone Calls

Now it's time to make some calls to the outside world. All you need is a broadband Internet connection and a commercial VoIP service provider that does PSTN termination. Some extras to consider -- though perhaps they are not so important for your test lab -- are migrating an existing phone number and 911 services. Not all providers offer these.

You want a "BYOD," or bring-your-own-device provider that is friendly to Asterisk, like this sampling of inexpensive, Asterisk-friendly providers:

- Broadvoice.com
- Nufone.net
- Quantumvoice
- VoIPJet
- TelaSIP

Every provider has their own Asterisk set-up instructions, so be sure to follow them because there is no generic configuration that works for all of them. When you configure Asterisk to use one of these providers, this is called setting up a new trunk. You'll need both an incoming and an outgoing trunk.

Firewall Configuration
To get through your firewall you'll need these ports forwarded to your Asterisk server:

4569 TCP/UDP
5004-5082 TCP/UDP
10000-20000 TCP/UDP

If you have a NAT firewall you must edit /etc/asterisk/sip.conf on the server, adding these lines:

```
externip = 1.2.3.4
localnet = 192.168.1.0/255.255.255.0
nat=yes
```

For "externip" use your own public IP, and "localnet" is your LAN. Be sure to check the instructions of your service provider for any special firewall configurations.

If you're not used to editing text files in the console, now is the time to learn, because even with Asterisk@Home you'll have to do this. Asterisk comes with both the vi and Nano text editors. Nano is easy to use. Open files like this:

nano/etc/asterisk/sip.conf

Basic commands are always displayed when Nano is open, so you'll learn your way around quickly.

Digital Receptionist
Your Digital Receptionist routes incoming calls, so the next step is to set up this feature.

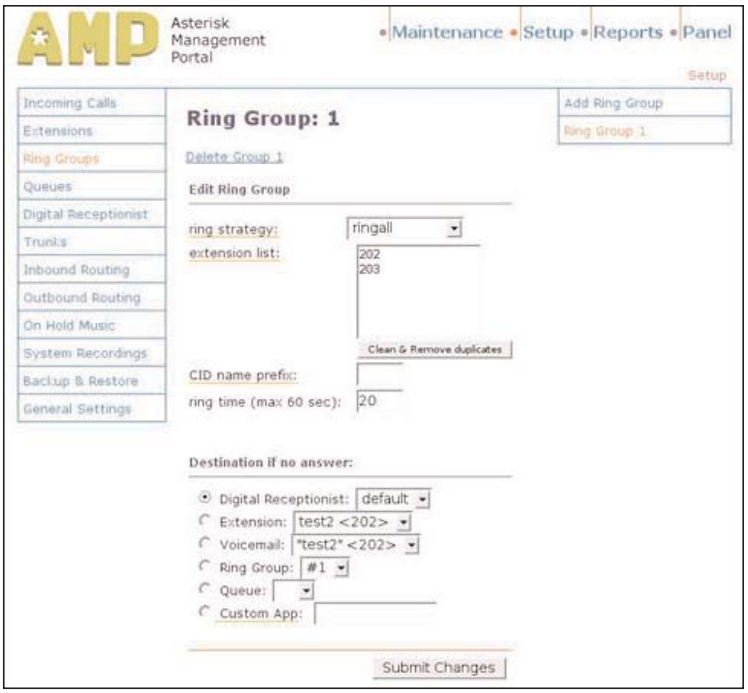
Open Setup --> Digital Receptionist. The first set-up window walks you through recording a greeting. The following windows are self-explanatory, and will walk you through setting up your various options. You may have several different Digital Receptionist menus, as Figure 4 shows.

Figure 4

Ring Groups

Setting up Ring Groups is optional. Some folks like to have all extensions ring on incoming calls. Asterisk can ring all extensions at once, or one at a time in sequence. Open Setup --> Ring Groups. Select the extensions you want in the group, like Figure 5 shows, and the action to take if no one answers.

Figure 5



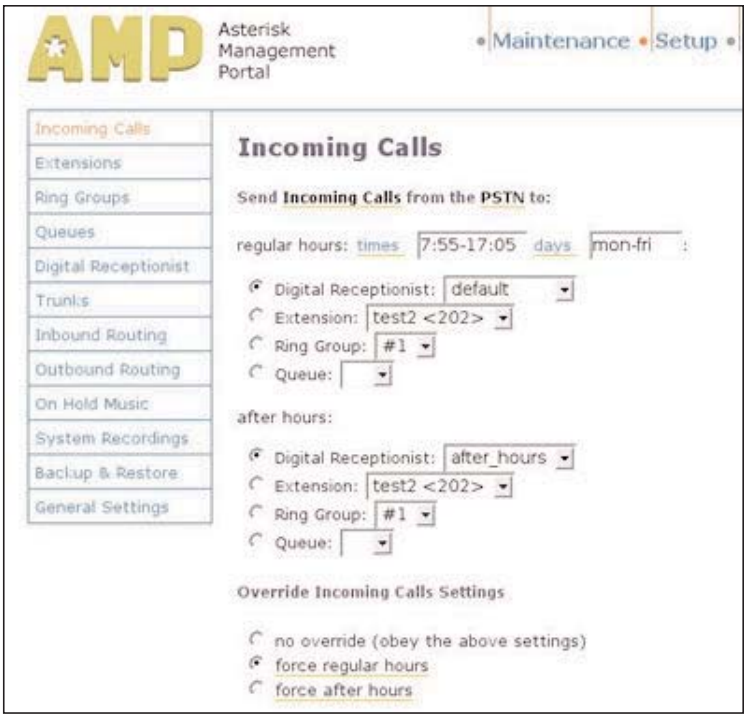
Incoming Calls

Now open Setup --> Incoming Calls. This controls how incoming calls from outside your network are handled at different days and times, as Figure 6 shows.

This is where you put your Digital Receptionists to work.

Now you can test just about any Asterisk function you can think of: different features, different hardware, do load-testing, and various networking tweaks and optimizations.

Figure 6



Securing Your Server

With our test lab up and running, it's time to lock down our Asterisk server, and that begins with secure passwords.

Asterisk@Home ships with a bunch of default passwords that many people know. Moreover, it sends server administration traffic in the clear, rather than over HTTPS. This means that anyone on your local network could easily sniff out all those passwords after you go through the trouble of changing them.

OpenSSH should be configured to use RSA key pairs instead of the root system login, which is both more secure and more convenient. Disconnect your Asterisk server from the network, and away we go.

Password Management

Strong passwords are fundamental defenses against intrusion. The world is chock-full of automated password crackers that crack easy passwords in seconds. Passwords should not be words, names, places, birthdates, Social Security numbers, or pet names. In other words, don't use anything that will be found in a dictionary or can be related to you in any way. Cracker dictionaries even include common misspellings. Random sequences of letters, numbers, and punctuation marks are best, no fewer than eight characters.

First we'll take care of the more important passwords and security holes.

CentOS Linux Password
The default login on your Asterisk@Home server is user "root"; the password is "password." This is the most important password of all, because this is the key to the kingdom. Log in on the command-line of the server and run the passwd command:

```
# passwd
Changing password for root
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

passwd is a standard Linux command. The rest of the password commands are Asterisk@Home commands.
Asterisk Management Portal Password

While you're still on the command line, run the passwd-maint script to change the password for the maint user, which controls AMP:

```
# passwd-maint
-----------------------------------------
Set password for AMP web GUI and maint GUI
User: maint
-----------------------------------------

New password:
Re-type new password:
Updating password for user maint
```

A related user is wwwuser which also has AMP access, except it is blocked from using the Maintenance tab. Change it with this command:

```
# passwd-amp
```

Disable Alt+F9
Hitting Alt+F9 on the Asterisk server bypasses the root login and takes you directly to the administration console, which does all the same things as AMP, but the graphics. You might leave this alone if you are confident in your physical security. Remember the ancient Unix security dictum: "Anyone with physical access to the box owns it." To disable it, do this:

```
# nano /usr/sbin/safe_asterisk
CONSOLE=no
```

Using the Nano Text Editor
The Nano text editor commands are displayed on the screen when you open it; to get more help hit ^G, which means the Control key plus the letter g, lowercase. Don't bother trying to make it a capital G, even though it is displayed that way. The Nano man page (*man nano*) may be helpful.

Just to keep it interesting, some commands do require using the Shift key, like the command to navigate to a specific line number, which is ^_ , or Control Shift Underscore.

Most businesses will look to integrating VoIP into their existing systems instead of totally replacing them. When comparing VoIP to standard PBX type phone systems, you soon begin to see some of VoIP's disadvantages.

For starters, when dealing with VoIP in high utilization scenarios, quality of service (QOS) assurances become difficult to deliver, compared to dealing with an old fashion PBX system. Quite often, the same scalability characteristics companies find attractive can ultimately be the reason their implementation of the technology initially fails.

High-end VoIP networks, such as those in large calling centers or a corporate headquarters with thousands of users, can become so complex that QOS level guarantees become harder to assure versus the traditional circuit switched voice network that has clear and concise capacity restrictions that built into the system and around which quality of service levels can easily be guaranteed and benchmarked.

VoIP does make physical provisioning and installation much easier versus a PBX installation, which requires a network of electrical wires, loops, and switches in order to function. A VoIP installation, on the other hand, will use your existing IP network so the logistics of building your VoIP network are largely simplified since the required physical elements are already in place.

The key advantage to standardizing your IP-based network for data, applications, and now VoIP, is that your administrators will have only one network to maintain. This means supporting only a single network cabling system, rather than separate systems, one for voice and one for data. And if you choose to move to WiFi Ethernet then you don't even need most of the cabling. We can compare this scenario to the old school PBX administrators that will still be required to maintain a separate local area cabling network for just the PBX system.

There is also the constant possibility of a virus infecting your network. If this happens to a standalone data network, then your employees can still make phone calls with the old school and isolated PBX network and continue data entry manually for a short time. However, when you combine these two networks, your VoIP phone calls may no longer be possible in this scenario.

-- Mike Houghton, EnterpriseITPlanet.com

Commands like "M-Y" mean Alt key plus y. M stands for Meta key. Why not just say Alt key? On old Sun systems the Meta was a key marked with a diamond, and on Macintosh it's the Command key. On modern systems some users prefer to use a custom keyboard mapping, so the Meta key is wherever they choose to put it. But for most of us, it's the Alt key.

ARI (Asterisk Recording Interface) Password

```
# nano -w
/var/www/html/recordings/includes/main.conf
```

On line 53, change the admin password within the quotes:

```
$ari_admin_password = "ari_password";
```

Hit ^w to search for "ari_password", or ^_ to go directly to line 53.

If you're thinking, "Um, storing passwords in plain text is not a good idea," you are correct. But that's the way it is for now, so guard your root password and Asterisk server well.

Flash Operator Panel (FOP) Password

Close out the /var/www/html/recordings/includes/main.conf file with ^X, then hit Y to save your changes. Then:

```
# nano -w /var/www/html/panel/op_server.cfg
```

Down near the end of the file, change the password on this line:

```
;security_code=passw0rd
```

MeetMe Password

Exit Nano and run this Asterisk@Home command:

```
# passwd meetme
```

System Mail Password

Use this command:

```
# passwd admin
```

A2Billing Password

Go to http://[your-Asterisk-IP]/a2billing and log in with "root" and "myroot." Go to Administrator í Show Administator to change both the default user passwords.

Sugar CRM Password
Click "CRM" on the Asterisk@Home splash page. Login with "admin" and "password" then click "My Account on the upper right to set a new password.

Next, we have to ensure that all Web administration traffic is encrypted, and we'll lock down OpenSSH more tightly.

Locking Down OpenSSH
By default, Asterisk@Home sets up OpenSSH to run after installation, and to accept root logins. Accepting remote root logins is not the best security practice, because it leaves the door open for brute-force attacks on the root account.

You might be thinking that you don't need to worry about these things because your Asterisk server is safely tucked behind your stout firewall, using a non-routable private IP. You are right that this reduces the potential for attacks from the Internet. However, should a remote attacker succeed in getting behind your firewall, it's better for them to find more barriers, rather than a wide-open welcome. And don't forget that most security breaches are inside jobs, rather than silly Hollywood-type break-ins from the outside.

There are a couple of different ways to make OpenSSH more secure. A simple way is to create an ordinary, unprivileged user on the Asterisk server, use this account for remote logins, then disable remote root logins. To set this up, log into the server from another PC on your LAN and create this user, using any name you like:

```
carla@windbag:~$ ssh root@192.168.1.25
Last login: Tue Apr 25 13:13:35 2006 from 192.168.1.10
Welcome to Asterisk@Home
--------------------------------------------
For access to the Asterisk@Home web GUI use this URL
http://
For help on Asterisk@Home commands you can use from this
command shell type help-aah.
```

```
[root@asterisk1 ~]# useradd freduser
[root@asterisk1 ~]# passwd freduser
Changing password for user freduser.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@asterisk1 ~]#
```

Now exit the root login then login as your new user:

```
[root@asterisk1 ~]# exit
Connection to 192.168.1.25 closed.
carla@windbag:~$ ssh freduser@192.168.1.25
```

After you are logged in, use the su (switch user) command to become root:

```
[freduser@asterisk1 ~]$ su
Password:
[root@asterisk1 freduser]#
```

Now open /etc/ssh/sshd_config, and add these lines:

```
[root@asterisk1 freduser]# nano /etc/ssh/sshd_config
```

```
PermitRootLogin No
AllowUsers freduser
Protocol 2
```

Then restart OpenSSH:

```
[root@asterisk1 freduser]# /etc/init.d/sshd restart
```

The AllowUsers directive is a nice way to preserve the flexibility of logging in from random remote hosts on your LAN, while blocking unauthorized users and brute-force attacks on the other Asterisk system accounts. OpenSSH supports two ssh protocols, 1 and 2; ssh1 is obsolete and weak, so it's important to limit your SSH sessions to Protocol 2 only. This makes SSH logins a two-step process, which is a bit inconvenient, but it adds a significant measure of security. Our little "freduser" has no power to do anything on the server, so even if an attacker succeeded in cracking freduser's account, the attacker would have to escalate to the root user to do any damage. This is called "privilege escalation." Privilege escalation is a fundamental tactic in any Linux intrusion attempt, because an attacker can't touch system files without root powers. This is why old Linux/Unix admins always nag about "don't do anything as root except what you really really have to." Strong passwords work, so make sure freduser has one.

Using Public Key Authentication

A second way to tighten up remote SSH access is to use public-key authentication. This protects your system passwords because you authenticate with a cryptographic key, instead of using a login/password. In addition to disabling root logins, you should also disable password authentication with this line in /etc/ssh/sshd_config:

```
PasswordAuthentication no
```

Now you can sit back and laugh at brute-force SSH attacks, because they simply won't work.

Why Remote Administration?

If you're wondering why you can't just sit down at your Asterisk server to do all your command-line chores, the answer is you can. So, if you don't need SSH access, you should turn it off entirely. Use the chkconfig command to do this:

```
# /sbin/chkconfig --del sshd
```

This doesn't turn off a running SSH session, but only prevents it from starting up at boot, so you need to shut it down:

```
# /etc/init.d/sshd stop
```

Securing AMP Traffic

Any server administration done over a Web interface is transmitted in cleartext, unless you enable HTTPS. HTTPS is SSL over HTTP; a nice easy way to encrypt HTTP traffic. To activate it on your Asterisk server all you need to do is install the Apache SSL module, then restart Apache (Apache is the Web server included in Asterisk@Home):

```
# yum -y install mod_ssl
# /etc/init.d/httpd restart
```

Then all you have to do is remember to point your Web browser to https://[asterisk-server].

This content was adapted from VoIPPlanet.com and written by Carla Schroder.

JupiterWeb eBooks bring together the best in technical information, ideas and coverage of important IT trends that help technology professionals build their knowledge and shape the future of their IT organizations. For more information and resources on networking, visit any of our category-leading sites:

www.enterprisenetworkingplanet.com
www.instantmessagingplanet.com
www.opticallynetworked.com
www.practicallynetworked.com
www.voipplanet.com
www.wi-fiplanet.com
www.opennetworkstoday.com
www.jupiterwebcasts.com/networking

For the latest live and on-demand Webcasts on networking, visit: www.jupiterwebcasts.com/networking