iPhones and Linux

[Some tips for preserving your sanity]

Hal Pomeranz
Deer Run Associates

Hal Pomeranz	hal@deer-run.com
Deer Run Associates	www.deer-run.com
PO Box 50638	(541)683-8680
Eugene, OR 97405	(541)683-8681 (fax)

In early November, 2008, Deer Run purchased Apple iPhones for it's Executive Officers (i.e., my wife Laura and me). Laura was really jazzed about her "cool new iPhone". My excitement was tempered by the fact that I knew it was going to take me several days of fiddling to make the darn phones work with our largely Linux-based infrastructure. Here's some notes about what I learned during this process…

Making the Phone "Work"

- Goals:
  - Calendar/contact synchronization
  - Join WPA2 network with TLS authentication
  - Access IMAP server for email, both at home and on the road
- Constraint: Has to be "wife-operable"
  - Should only use built-in interfaces
  - No jailbreaking to use SSH, or similar hacks

My goals were actually pretty modest:

• We need the phones to be able to synchronize their calendars and contact settings to our personal systems (Windows for Laura, Ubuntu Linux for me).

• We wanted the phones to be able to use our home/business Wi-Fi network. The trick here is that my "home" network uses WPA2/Enterprise security with TLS (certificate) authentication, just like my grownup enterprise customers. I'd already done the research to learn that the iPhone 3G was capable of this, I just didn't know how yet.

• Our mail server lives on our "home" network, rather than being hosted someplace else. It's a CentOS Linux box running the Dovecot IMAP server. IMAP access is only via SSL. We wanted to be able to retrieve email not only when we were on our "home" network, but also while we were traveling.

The major constraint was that whatever solution(s) I came up with had to be usable by Laura. Now Laura's an extremely sophisticated computer user, but I know from previous experience that there's a limit to what she's prepared to tolerate in terms of "user experience". For example, when I set up some SSH sessions for her on her laptop that had custom tunnels configured so that she could access her mail remotely with Thunderbird by simply changing her IMAP server settings, she decided it was easier to just SSH into the mail server and read her email from the terminal with mutt when she was on the road (I love my wife, she rocks the command line). Basically, Laura wanted to use the standard phone interfaces with no extra mucking around– it was up to me to do the required magic to make everything "just work".

Calendar/Contact Sync

- The only way to sync to Evolution appears to involve sync to external site first
- That is so not happening for me…
- So I'm running iTunes/Outlook in VMware [yes, I punted this one, sorry]
- Btw, *DO NOT* attempt to upgrade your iPhone software from VMware!

To be honest, I was dreading getting the iPhone to sync with my wife's Outlook installation– I figured it would only sync with iCal on the Mac or some such nonsense. Turns out that I needn't have worried: recent versions of iTunes sync contacts/calendars to Outlook on Windows with absolutely no fuss. Win!

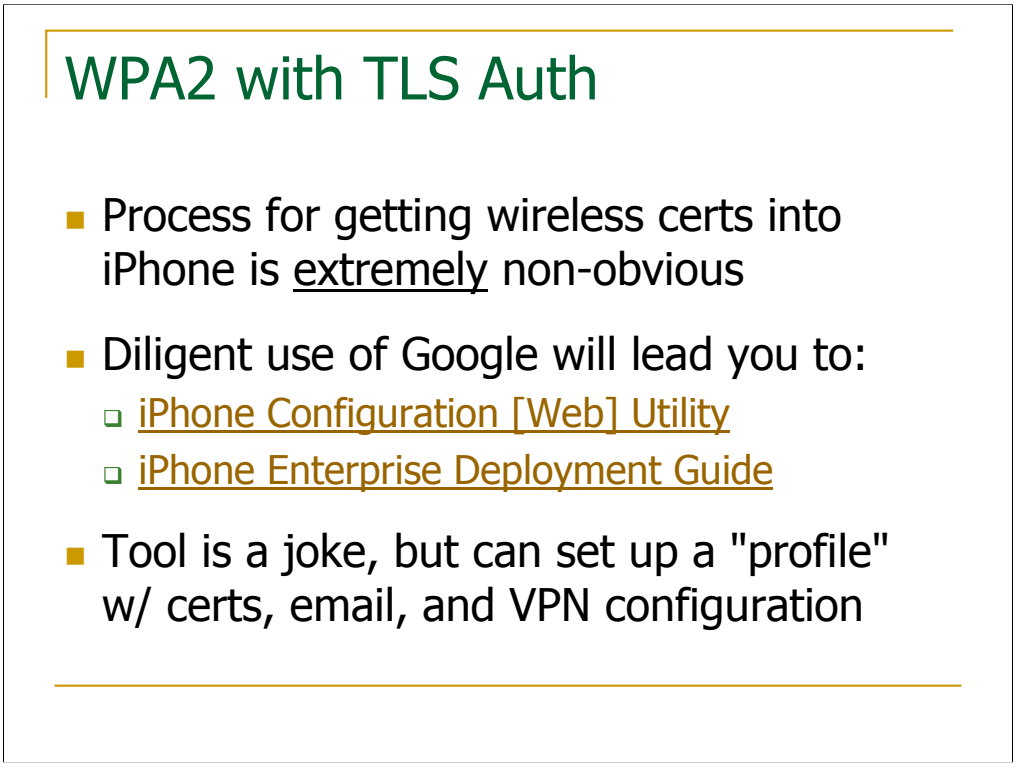
The Linux situation, however, is not so happy. There doesn't appear to be a freely-available tool for directly connecting the iPhone to Evolution or an equivalent tool on the Linux desktop. The only solutions I could find involved syncing the phone to an external Internet service, exporting the data into some other format, and then importing the data into Evolution. Not only was the process ridiculously convoluted, it meant I'd have to store my calendar and contact data on some random web site. I'm a paranoid security consultant and that's a non-starter for me.

So I'm running Outlook in VMware so that I can back up the data on my iPhone. Sorry Open Source fans, but it's the best that I could do. I'll note that if you're using the free VMware server product like I am, you'll need to upgrade to one of the 2.x versions in order to get support for USB 2.0, which is required to talk to your iPhone.

Frankly, you also need VMware to run iTunes (Wine support for the latest versions of iTunes just isn't there yet). iTunes inside of VMware "works" pretty well. However, my attempt to upgrade the software in my iPhone from VMware was a disaster– I nearly bricked the phone. The problem is that that software update process drops/raises the USB connection at several points, each time requiring that you reconnect the device to the VMware image… manually. Unfortunately, while you're clicking around trying to get the phone reconnected, iTunes bombs out the upgrade leaving you with a hosed phone.

I was able to recover by doing a phone OS restore from my wife's Windows machine, followed by restoring my data from my VMware image. However (and curse you Apple for this!) iTunes apparently doesn't back up many of your personalization settings and I had to re-enter those manually. It sucked.

At this point, I only do software upgrades on the phone from Laura's Windows machine. Another failure for the Open Source platform…



As I mentioned earlier, I confirmed from several sources on the web that the iPhone 3G could handle WPA2/Enterprise with TLS auth– this was in fact one of my technical criteria for the next phone I bought (my old Windows Mobile based phone couldn't deal with this and it was annoying). But once I sat down with my new iPhone, I couldn't see any obvious way to load the necessary certificates into the darn phone. That's because there isn't one– at least not one that comes with the phone.

Once I finally got around to Surfing The Fine Web, I discovered that this is an "enterprise" feature, and there's a special "iPhone Configuration Utility" (freely available) to allow enterprise administrators to create iPhone "profiles" that can be loaded into users' phones. Profiles can contain not only information like digital certificates, but also set up email and VPN account parameters, phone passcode settings (i.e., requiring users have one), etc.

The iPhone Configuration Utility comes in a couple of different versions: there's a MacOS native client and a "web-based" utility that runs on either MacOS or Windows (Why does a "web-based" tool require a specific OS platform? More on that in a minute). Download the software from:

http://www.apple.com/support/iphone/enterprise/

You'll also want the "iPhone Enterprise Deployment Guide" document that describes how to set up and use the tool:

http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf

What's Wrong With This Picture?

- Tool login: "admin", password: "admin"
- Load certs and other config data
- Save XML profile (certs password-protected, user prompted when profile loaded)
- User loads profile via email message or via Web page (small hack required for Apache)
- Repeat for each phone/user
- Btw, can only have one profile per phone

Now if Apple seriously believes that their iPhone Configuration Utility is an "enterprise" grade tool, then they're delusional. Since I don't have any MacOS boxes on my network, I downloaded the Windows version of their "web-based" utility. What you get is a Rails app coupled with a small web server: once you run the installer you can connect to localhost:3000 to access the app. Upon connecting to that address, you're presented with a username/password prompt. The "Enterprise Deployment Guide" cheerfully informs you that you log in as "admin" with the password "admin". As far as I can tell, there's no way to change this default login without hacking the underlying code. Seriously guys, why even bother with the login if you're going to be like this?

Once you "log in" to the app, you're presented with tabs for entering different types of information (Wi-Fi settings, VPN, email, certificates, etc). Initially I thought that I could create a profile that just had Wi-Fi settings in it, and then later I could go back and add additional profiles to my phone that handled email and VPN settings. But it turns out that the iPhone can only have one profile loaded into the phone at a time, and loading a new profile clobbers all the settings in the old one. So you need to create a comprehensive profile that covers all the settings for a particular user. I also have to believe that this "one profile at a time" behavior is a serious PITA for folks who migrate between networks a lot, though I'm sure it makes the IT managers at each of those individual networks sleep a lot better at night.

So you create a profile for a user by filling in the information on the various tabs. You'll need to copy the certificate files for the user (as well as the root if you're using self-signed certs) to the machine where you're running the app so that you can load them via your web browser. You'll probably want to password-protect the certs so that they can only be unlocked by the user of the phone. This happens at the moment the user loads the profile– in fact there's a whole lot of passwords requested during the profile loading process, including email and VPN passwords along with the certificate passwords.

Once you've got the information filled out in the app, you "export" the profile. The "export" function dumps the profile out in an XML format. Frankly, in a real "enterprise" situation, I'd use the Configuration Utility to dump out a single profile as an example and then reverse-engineer the XML so that I could write a Perl script to crank these things out by the truckload for all of my users.

Once you've got the XML file, how do you get it into the phone? Apple currently gives you two choices, neither of which is particularly great. The first is to send it to the user as an email attachment– when the user opens the email on their iPhone, the phone is supposed to recognize the attachment type and prompt the user to load the profile. Of course, if you're relying on the profile to configure the user's email settings, you've got a bit of a chicken and egg problem here. Also, I'm just waiting for the first iPhone attacks that send malicious profiles to unsuspecting users via email, but that's another story.

The second approach for loading profiles is to have users download the profiles via the Safari web browser on their phones. Again you have a potential "chicken and egg" problem here– what if the only way users can access the web server holding the profiles is to load TLS certs onto their phone to access the local Wi-Fi? Seriously Apple, how hard would it have been to also give users a mechanism to download their initial profiles via a direct USB connection? Idiots.

In my case, I set up an isolated web server on an air-gapped network with an open access point so that I could download initial profiles into the phone. For "enterprise" deployments I guess you'd have to set up some sort of kiosk to handle this, or use iPhone developer tools to load the profiles. Yuck!

By the way, if you're exporting the profiles from an Apache web server, you'll need to update your web server's `/etc/mime.types` file as follows:

```
# For iPhone mobile device profiles
application/x-apple-aspen-config        mobileconfig
```

This setting associates the file extension `.mobileconfig` with the appropriate MIME type so that Apache will send the right `Content-type` header to Safari. You will need to restart your web server after making the above change.

So How To Deal With Email?

- On "home" network, email access is straightforward: iPhone talks IMAP/SSL
- For external access, decided to put phone onto "home" network via VPN:
  - VPN choices are Cisco IPSEC, L2TP, PPTP
  - If you have a Cisco VPN concentrator, great!
  - Otherwise, grab yourself a copy of Poptop (Open Source PPTP server)

Another selling point of the iPhone from my perspective is that you can use open protocols like IMAP to fetch your email (as opposed to, say, all of the proprietary crap you have to wade through to make your Blackberry function). Once I got my iPhone talking on our "home" Wi-Fi network, it was easy to have it fetch email from our Dovecot server via SSL.

The question then became, what to do about email access when we were traveling? I considered various proxy ideas, but they would all involve changing the IMAP settings in the phone whenever we were somewhere other than our home network and then switching back once we returned to home base. This seemed like a non-starter from Laura's perspective.

So how do you make the phone think it's on the "home" network when you're somewhere else in the world? Why with a VPN of course! Unfortunately, the iPhone VPN options are somewhat limited at this point. If you have a Cisco VPN concentrator, the iPhone has Cisco VPN client support bundled in, allowing you to do IPSEC. However, the IPSEC implementation in the iPhone is Cisco-specific and won't work with my OpenBSD firewall's IPSEC implementation for example. Again, how hard would it have been to put more general IPSEC support into the phone if they were going to the trouble of putting the Cisco hacks in?

Since I wasn't about to head out and buy a Cisco VPN concentrator for my home-office network, PPTP seemed like the only option that I could implement in an Open Source infrastructure (doesn't seem like L2TP is an option here). The Open Source PPTP implementation is Poptop (*http://poptop.org/*)

Some Poptop Notes

- I wasted several days trying to get Poptop working on OpenBSD (ultimately failed)
- Poptop really wants to run on Linux, support community is Linux-focused
- To get it working with iPhones:
  - Basic Poptop setup from recipe on web
  - There were a couple of iPhone-specific hacks that I had to work out on my own

Initially I really wanted to run Poptop on the OpenBSD server I use as a firewall for my home network. I wasted several days trying to make this work but finally gave up in disgust. Poptop uses the native pppd in whatever operating system its running in, and for whatever reason Poptop doesn't get along with the BSD pppd. The debugging output from the BSD pppd is just about useless– mostly because there doesn't seem to be a document that you can use to decipher what the debugging messages are telling you. It's maddening really.

So finally I punted and set up a CentOS box behind my firewall to run Poptop. This meant setting up some inbound NAT rules on my firewall to get the necessary packets to the right place, but it's pretty straightforward. PPTP uses 1723/tcp for authentication and GRE for the actual VPN traffic, so as long as you pass those through to your VPN server, you should be OK. Some notes on what I did to get Poptop to build on OpenBSD and my firewall configs can be found in these mailing list threads:

http://www.nabble.com/Proxy-ARP-issue--td20611872.html

http://www.nabble.com/poptop-(1.3.0)-on-OpenBSD-4.4-td21264232.html

For getting Poptop set up on CentOS, the following blog posting was very useful:

http://www.sharedknowhow.com/2008/09/linux-vpn-server-installation-for-use-with-iphone/

As I noted in one of my mailing list threads, in addition to the instructions in the above blog posting, you also need to set "`noipv6`" and "`noccp`" in `/etc/ppp/options` in order to make it possible for the iPhone to connect.

All That Being Said…

- The iPhone is *sexy*…
- Best phone I've ever owned (for my needs)
- Some useful apps:
  - TouchTerm (free SSH client)
  - mSecure (password locker)
  - Pandora Radio (buh-bye XM subscription!)
  - TwitterFon (meh, but good basic Twitter app)

Despite all of the trouble I had getting things working, and despite some of the idiocy perpetrated by Apple here, Laura and I really do love our iPhones. It does everything I need it to do with no hassles (other than the initial infrastructure setup issues detailed here). For trips where I don't need to teach a class or give some sort of talk, the iPhone is functional enough that I can leave my laptop at home. I can't express how liberating it is to not have to lug a laptop through airport security. And the iPhone is *soooo* much cooler than a Blackberry…

There are a few other apps that I've found useful for the iPhone:

• I can't live without an SSH client. Obviously there are some platform limitations (screen real estate, lack of keyboard) that make the iPhone less than ideal for terminal-based applications, but TouchTerm is an excellent free app that deals with these limitations about as well as I could imagine can be done.

• I also need a secure password keeper app. After doing some research, I actually paid a few dollars for the mSecure app and have been happy with it. Customer support is very responsive– I think the support email alias goes to the developer of the app and he personally answers the emails. Can't beat that.

• I've become mildly addicted to Pandora Internet radio, and there's a free iPhone client. With a simple cassette adaptor, I'm able to play the music in my car and that's enabled me to jettison my XM radio subscription (though I'm hoping MLB will start making the radio feeds from baseball games available via their MLB On-Deck app, because otherwise I might go into withdrawal).

That's It!

- Thanks for listening!
- Any final questions?

This space intentionally left blank…